
175 PTAS
RICARDO STANICIC
mi computer
75
CURSO PRACTICO DEL ORDENADOR PERSONAL,
EL MICRO Y EL MINIORDENADOR
apricot PORTABLE
apricot
VOICE
STOP
SHIFT
9 788485 822836 100075
Editorial Delta, S.A.

MI COMPUTER CURSO PRACTICO DEL ORDENADOR PERSONAL, EL MICRO Y EL MINIORDENADOR

Publicado por Editorial Delta, S.A., Barcelona

Volumen VII-Fascículo 75

Director:	José Mas Godayol
Director editorial:	Gerardo Romero
Jefe de redacción:	Pablo Parra
Coordinación editorial:	Jaime Mardones Francisco Martín
Asesor técnico:	Ramón Cervelló

Redactores y colaboradores: G. Jefferson, R. Ford, F. Martín, S. Tarditti, A. Cuevas, F. Blasco
Para la edición inglesa: R. Pawson (editor), D. Tebbutt (consultant editor), C. Cooper (executive editor), D. Whelan (art editor), Bunch Partworks Ltd. (proyecto y realización)

Realización gráfica: Luis F. Balaguer

Redacción y administración:
Paseo de Gracia, 88, 5.°, 08008 Barcelona
Tels. (93) 215 10 32 / (93) 215 10 50 - Télex 97848 EDLTE

MI COMPUTER, *Curso práctico del ordenador personal, el micro y el miniordenador*, se publica en forma de 96 fascículos de aparición semanal, encuadernables en ocho volúmenes. Cada fascículo consta de 20 páginas interiores y sus correspondientes cubiertas. Con el fascículo que completa cada uno de los volúmenes, se ponen a la venta las tapas para su encuadernación.
El editor se reserva el derecho de modificar el precio de venta del fascículo en el transcurso de la obra, si las circunstancias del mercado así lo exigieran.

ISBN: 84-85822-83-8 (fascículo) 84-7598-067-2 (tomo 7)
84-85822-82-X (obra completa)
Depósito Legal: B. 52-84

Fotocomposición: Tecfa, S.A., Pedro IV, 160, Barcelona-5
Impresión: Cayfosa, Santa Perpètua de Mogoda (Barcelona) 198506
Impreso en España-Printed in Spain-Junio 1985

Editorial Delta, S.A., garantiza la publicación de todos los fascículos que componen esta obra.

Distribuye para España: Marco Ibérica, Distribución de Ediciones, S.A., Carretera de Irún, km 13,350. Variante de Fuencarral, 28034 Madrid.
Distribuye para Colombia: Distribuidoras Unidas, Ltda., Transversal 93, n.° 52-03, Bogotá D.E.
Distribuye para México: Distribuidora Intermex, S.A., Lucio blanco, n.° 435, Col. San Juan Tlihuaca, Azcapotzalco, 02400, México D.F.
Distribuye para Venezuela: Distribuidora Continental, S.A., Edificio Bloque Dearmas, final Avda. San Martín con final Avda. La Paz, Caracas 1010.

Pida a su proveedor habitual que le reserve un ejemplar de MI COMPUTER. Comprando su fascículo todas las semanas y en el mismo quiosco o librería, Vd. conseguirá un servicio más rápido, pues nos permite realizar la distribución a los puntos de venta con la mayor precisión.

Servicio de suscripciones y atrasados (sólo para España)

Las condiciones de suscripción a la obra completa (96 fascículos más las tapas, guardas y transferibles para la confección de los 8 volúmenes) son las siguientes:

a) Un pago único anticipado de 19 425 ptas. o bien 8 pagos trimestrales anticipados y consecutivos de 2 429 ptas. (sin gastos de envío).

b) Los pagos pueden hacerse efectivos mediante ingreso en la cuenta 6.850.277 de la Caja Postal de Ahorros y remitiendo a continuación el resguardo o su fotocopia a Editorial Delta, S.A. (Paseo de Gracia, 88, 5.°, 08008 Barcelona), o también con talón bancario remitido a la misma dirección.

c) Se realizará un envío cada 12 semanas, compuesto de 12 fascículos y las tapas para encuadernarlos.

Los fascículos atrasados pueden adquirirse en el quiosco o librería habitual. También pueden recibirse por correo, con incremento del coste de envío, haciendo llegar su importe a Editorial Delta, S.A., en la forma establecida en el apartado b).

Para cualquier aclaración, telefonear al (93) 215 75 21.

No se efectúan envíos contra reembolso.



Micros en la escuela

Iniciamos una serie en que analizaremos el impacto de los ordenadores en el ámbito de la educación

Tony Lodge

La versatilidad de una máquina
En la clase de hoy en día, un ordenador puede, con el software adecuado, reemplazar fácilmente la pizarra, el proyector de diapositivas, el televisor, el aparato para cassettes y la biblioteca. Donde no es posible la sustitución directa (el "rincón de pintura" de un aula de primaria, p. ej.) los ordenadores pueden complementar útilmente los métodos de enseñanza tradicionales. Y, lo más importante, el maestro queda liberado de tareas muy caras en cuanto a tiempo (marcar textos, etc.) y dispone de más tiempo para interactuar con cada estudiante sobre una base individual

"A largo plazo esperamos que los sistemas de información relacionados con el ordenador tengan tanta influencia en la educación como la que tuvo en su día el desarrollo del libro impreso." Estas palabras, extraídas de un informe del gobierno británico en 1978, las ha repetido recientemente el profesor Tom Stonier, una destacada autoridad en nueva tecnología. Tanto él como otras personas, incluyendo al experto en educación norteamericano Seymour Papert, creen que las escuelas, tal como las conocemos en hoy en día, desaparecerán durante el curso de nuestra vida.

Esta visión más bien radical es una consecuencia natural del nivel de compromiso de estos pedagogos en el campo de la educación informatizada. En la práctica, no obstante, el cambio se está produciendo con gran lentitud y conducido por los educadores y no por las máquinas que tienen a su disposición. Sin embargo, son muchas las personas, tanto pertenecientes al sistema educativo como ajenas al mismo, que piensan que, aplicada adecuadamente, la informática puede causar un impacto significativo en el proceso del aprendizaje.

El empleo del ordenador en la clase se remonta a diversos proyectos y experimentos educativos realizados en el transcurso de los últimos veinte años, si bien algunos de los principios implicados han estado presentes desde mucho tiempo antes. En los años veinte se construyeron máquinas para enseñar simples, si bien no se popularizaron hasta los años cincuenta, cuando B. F. Skinner creó una primitiva máquina de enseñanza por realimentación. Skinner, uno de los psicólogos educativos que más influyeron en los años cincuenta y sesenta, desarrolló sus ideas a lo largo de dos décadas, experimentando con ratas y palomas. Uno de sus proyectos más inusuales involucraba un mecanismo autodireccional para un misil que llevaba tres palomas en su parte delantera, anticipándose de este modo en veinte años al Cruise. Skinner había condicionado a

los pájaros para que picotearan un punto del mapa recompensándolos con alimento. Cabe agregar que el proyecto fue rechazado por el Pentágono, lo que hizo que dejara tras de sí "una enorme cantidad de equipo inútil y unas docenas de palomas con un extraño interés por una zona de la costa de Nueva Jersey".

Las ideas de Skinner acerca del aprendizaje, sin embargo, fueron mejor recibidas. Abogó por un sistema de "programación lineal" (nada que ver con la programación de ordenadores) que implicaba presentarle información a un estudiante en pequeñas dosis cada vez y reforzando positivamente una respuesta correcta. En realidad, Skinner aplicaba a los estudiantes el mismo principio que había ensayado en sus "palomas piloto".

Programación lineal

De la programación lineal surgieron dos conceptos importantes: el *feedback* o realimentación y la *individualización*. Realimentación alude a la respuesta inmediata que se le da al estudiante al decirle si la respuesta dada es correcta o incorrecta. Lamentablemente, las máquinas de aprendizaje de Skinner, cuya operatoria se basaba en manivelas y botones a presión, sólo podía aceptar respuestas específicas. Individualización alude a la naturaleza directa del método, que permite que el estudiante trabaje a su propio paso y no al que le imponen sus compañeros de clase o su maestro.

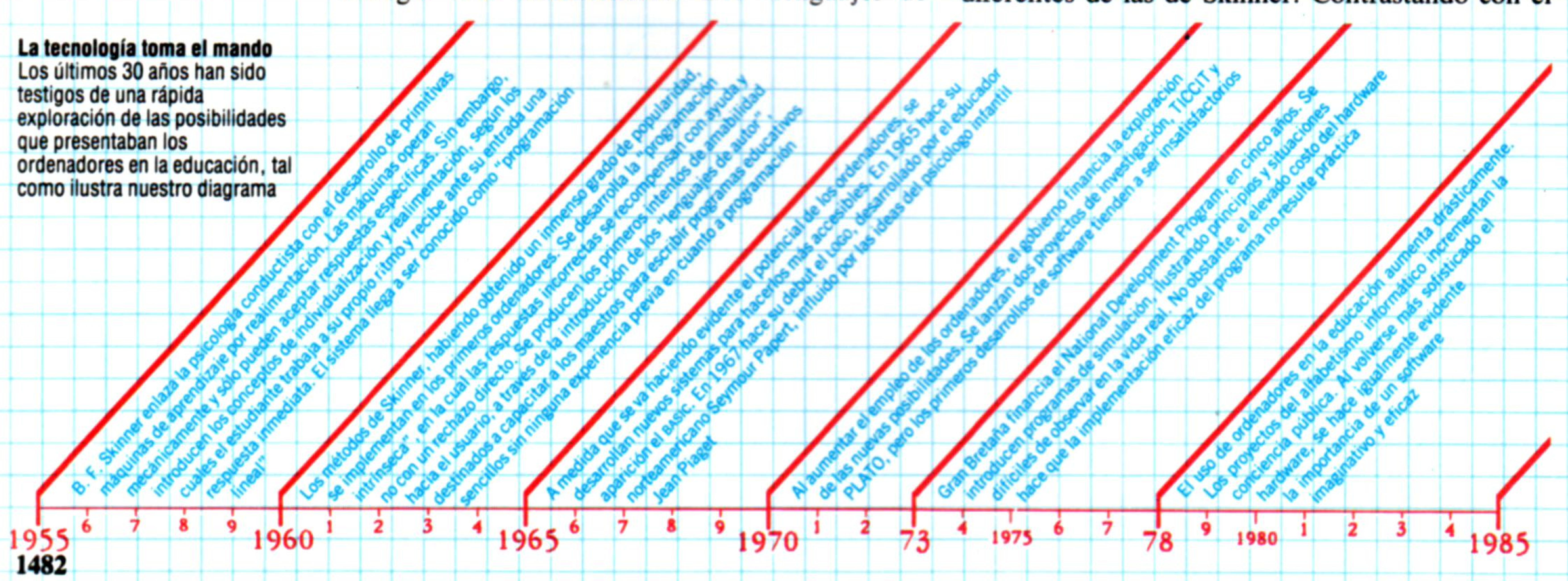
Los ordenadores se volvieron algo común cuando las máquinas de aprendizaje de Skinner se hallaban en la cima de su popularidad, y pronto se implementaron en los mismos los sistemas de la programación lineal. A pesar de las importantes limitaciones de este método, aún nos encontramos frecuentemente con programas que lo utilizan. "Lo siento, prueba otra vez" o "Bien hecho, lo has conseguido" aparecen centelleando en la pantalla, reforzando las respuestas a un alud de preguntas.

Un desarrollo de la programación lineal fue el *programa ramificado*. En vez de sólo obtener otro intento ante una respuesta incorrecta, el estudiante recibía ayuda y era luego reexaminado. El enfoque parece lógico, pero la "programación intrínseca" (como se la conoció) molestó en gran medida a Skinner y a sus discípulos. Sin embargo, para el estudiante la realimentación era mayor y estaba elaborada de forma más individualizada que antes, y corregía todo malentendido. Los "lenguajes de autor" se idearon para ayudar a los maestros a escribir este tipo de programas. Representaron un temprano intento por llegar a la "amabilidad" con el usuario, capacitando supuestamente a un maestro para escribir un programa educativo sin tener que aprender programación de ordenadores.

Tanto los programas lineales como los intrínsecos poseen graves inconvenientes. Consideran al estudiante como un recipiente vacío que hay que llenar con conocimiento, y ven el aprendizaje como la adquisición de hechos en lugar de experiencias. No dejan lugar a la autoexpresión, la imaginación ni la creatividad, y fomentan una única respuesta "correcta". Estos sistemas, que nos dan el término *aprendizaje programado*, han creado injustamente una mala reputación a todo el aprendizaje asistido por ordenador.

A pesar de la influencia restrictiva de las primeras máquinas para enseñar, se reconoció enseguida que los ordenadores ofrecían posibilidades totalmente nuevas para la educación. Gran parte de los primeros trabajos realizados en este ámbito han tenido una importancia perdurable y no sólo para los maestros de escuela. En 1965, en el Dartmouth College de Estados Unidos se preparó un proyecto para idear un lenguaje de enseñanza "lo más parecido posible al inglés" que les permitiera a los científicos, ingenieros y estudiantes escribir sus propios programas. Debido a las limitaciones de hardware de aquella época, cuando una máquina de 16 K ocupaba toda una sala de estar, hubo de diseñarse el lenguaje de modo que ocupara el menor espacio de memoria posible. El resultado fue el Beginner's All-purpose Symbolic Instruction Code (código simbólico de instrucciones para fines generales destinado al principiante), o BASIC, y ha permanecido con nosotros desde entonces.

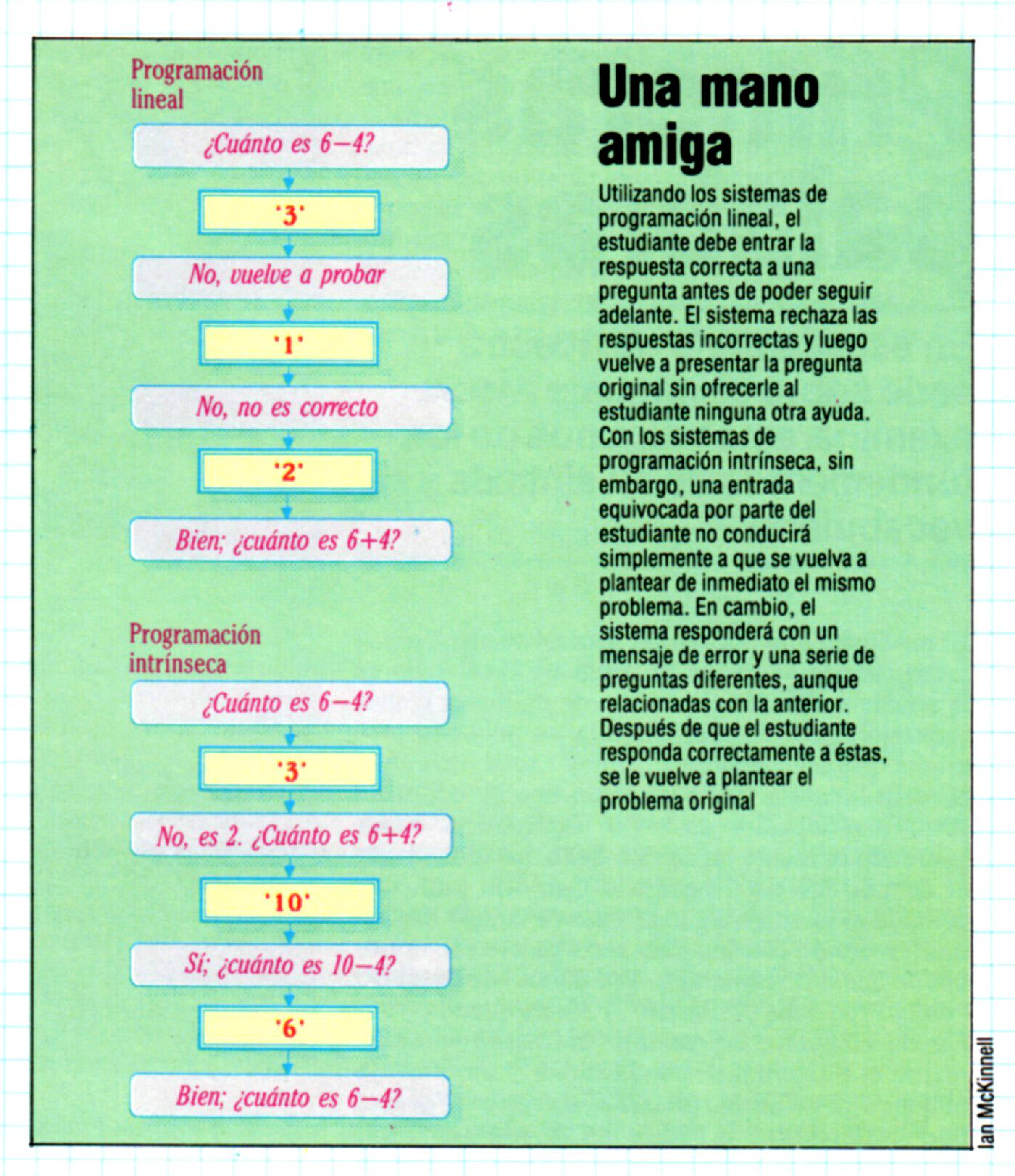
Un par de años después de que se creara el BASIC, un equipo de científicos en ordenadores y especialistas de la educación del Massachusetts Institute of Technology, idearon un lenguaje de programación "diseñado para facilitar las cosas al programador en vez de al ordenador". El objetivo fue lograr que los niños pequeños controlaran un ordenador, una idea revolucionaria para aquella época, dado que todos los proyectos previos habían estado dirigidos a estudiantes universitarios. El jefe del proyecto era Seymour Papert, quien había estudiado con el psicólogo infantil Jean Piaget. Las ideas de Piaget acerca de la educación eran radicalmente diferentes de las de Skinner. Contrastando con el



La tecnología toma el mando
Los últimos 30 años han sido testigos de una rápida exploración de las posibilidades que presentaban los ordenadores en la educación, tal como ilustra nuestro diagrama

enfoque mecanicista de este último, Papert creía en la creación de un entorno en el cual el niño pudiera aprender a través del descubrimiento. Propuso que: "En vez de dejar que el ordenador programe al niño, debemos permitir que el niño programe al ordenador." El resultado de su trabajo fue el LOGO, un lenguaje que recientemente ha llegado hasta los micros y que se está popularizando en las escuelas primarias. El advenimiento del BASIC y del LOGO puso la programación de ordenadores al alcance de los estudiantes, incluso de los de nivel primario.

A pesar de los esfuerzos que se estaban dedicando al desarrollo de las técnicas de aprendizaje asistidas por ordenador, muchas personas seguían albergando dudas sobre los beneficios del empleo de la informática en el campo de la educación. En un esfuerzo por despejar parte de esta incertidumbre, la National Science Foundation de Estados Unidos decidió en 1970 invertir 10 millones de dólares a lo largo de cinco años para investigar al tema. Se crearon dos proyectos: TICCIT y PLATO.



Una mano amiga

Utilizando los sistemas de programación lineal, el estudiante debe entrar la respuesta correcta a una pregunta antes de poder seguir adelante. El sistema rechaza las respuestas incorrectas y luego vuelve a presentar la pregunta original sin ofrecerle al estudiante ninguna otra ayuda. Con los sistemas de programación intrínseca, sin embargo, una entrada equivocada por parte del estudiante no conducirá simplemente a que se vuelva a plantear de inmediato el mismo problema. En cambio, el sistema responderá con un mensaje de error y una serie de preguntas diferentes, aunque relacionadas con la anterior. Después de que el estudiante responda correctamente a éstas, se le vuelve a plantear el problema original

TICCIT y PLATO

El TICCIT (Time-shared Interactive Computer Controlled Information Television) tuvo como objetivo "proporcionar mejor instrucción a menor costo que la instrucción tradicional en las escuelas de la comunidad". Se le encargó la producción de hardware y software para educación a una empresa, que desarrolló un teclado para estudiantes que incluía mandos especiales de "control para principiantes", una visualización de televisión para texto, gráficos y videos, y un altavoz para mensajes sonoros. Éstos se configuraron en sistemas de 128 terminales, controlados por dos miniordenadores. Hubo problemas con el software y muchos estudiantes no finalizaron el curso, pero quienes lo terminaron obtuvieron mejores calificaciones que aquellos que habían estudiado con métodos convencionales. El TICCIT ha seguido utilizándose en los dos colegios piloto, pero no ha sido adoptado en otro sitio.

PLATO (Programmed Logic for Automatic Teaching Operation) consiste en una red de alrededor de 1 000 terminales conectados a un ordenador central en Illinois. Los terminales se pueden comunicar entre sí y acceder a una biblioteca de lecciones retenida en el ordenador principal. Las unidades de visualización de PLATO poseen paneles transparentes que permiten proyectar diapositivas en color sobre gráficos de ordenador. Un panel al tacto permite a los estudiantes comunicarse con el programa tocando la pantalla. Los maestros pueden escribir sus propios programas en un lenguaje de autor denominado TUTOR. Al igual que el TICCIT, la preparación del software consumió más tiempo y esfuerzo de lo previsto. Los resultados de los estudiantes que utilizaron el PLATO no fueron ni mejores ni peores que los de los estudiantes que emplearon los métodos convencionales, aunque a los que lo aplicaron el sistema les resultó más amable que a los que trabajaron con el TICCIT.

En 1973 el gobierno británico patrocinó un National Development Program para aprendizaje asistido por ordenador, de cinco años de duración y con un presupuesto de 2,5 millones de libras. La mayor parte del dinero se gastó en desarrollar software en forma de simulaciones, utilizándose el ordenador para crear situaciones difíciles o imposibles de observar en la vida real. Un programa, por ejemplo, simulaba la concentración de tintura en el torrente sanguíneo a diversos intervalos de tiempo y otros ilustraban reacciones químicas. No obstante, la mayor parte del software era muy poco imaginativo y el elevado costo de los ordenadores en los años setenta hizo que su uso no fuera práctico.

Todo sistema educativo sufre la influencia de la política, la economía y las demandas de la sociedad. Los ordenadores en la educación no sólo tienen que pugnar contra estos factores, sino también contra las presiones del cambio tecnológico. A ello se debe añadir el conservadurismo de quienes ven a los ordenadores como la simple representación de una extensión de las nada deseables técnicas de aprendizaje programado que analizáramos con anterioridad.

Mientras tanto, al igual que en todas las áreas de la informática, las tendencias dominantes de hoy en día en la informática educacional son consecuencia de la constante caída del precio de la memoria. Hoy en una escuela primaria de Londres hay más potencia informática que en todas las universidades británicas de hace 20 años. Un efecto de ello, según algunos críticos, ha sido la concentración de adquisición de hardware al costo del desarrollo del software. Sin embargo, ahora los políticos parecen más preparados para reconocer el software como un recurso nacional de importancia vital y, en la medida en que el precio del hardware continúe disminuyendo, podemos esperar ver cambios radicales en los sistemas educativos en los próximos diez años.

Primeras palabras

En este capítulo de nuestra serie sobre el lenguaje PASCAL examinaremos algunos de los fundamentos de su sintaxis y su vocabulario



Compilación popular
Una implementación profesional completa del PASCAL puede alcanzar un precio que supere al de un ordenador personal, pero recientemente se ha producido una ola de compiladores de precio razonable para micros personales populares. Mostramos una selección de los paquetes disponibles en la actualidad

El problema que por norma general se plantea primero cuando se utiliza un lenguaje compilado es aprender a abordar el proceso de múltiples etapas requerido para obtener siquiera un pequeño programa que sea ejecutable. Primero se ha de entrar el texto fuente a través de algún tipo de editor de texto o procesador de texto. Luego, tras haber guardado el fuente en cinta o disco, se debe cargar el compilador del lenguaje e instruirlo para que compile el fuente a alguna forma de código máquina (a menudo con una compleja cadena de opciones de líneas de comando). Por último, este archivo "objeto" se debe "relocatar" y enlazar con las rutinas de biblioteca en *run-time* necesarias. Posiblemente el programa se pueda cargar y ejecutar sin ningún otro esfuerzo, pero si el compilador genera un "seudocódigo" o código intermedio, entonces para ejecutar el programa se habrá de utilizar un intérprete en *run-time*.

Si todo esto le suena excesivamente complicado, le tranquilizará el hecho de que todos los paquetes de PASCAL disponibles para ordenadores personales evitan en gran medida estos problemas. Al menos durante el desarrollo del programa, su texto fuente, compilador y programa objeto pueden estar residentes en la memoria. Ello es posible gracias a la eficacia y el pequeño tamaño del PASCAL, y el sistema resultante con frecuencia no es más complicado, al utilizarlo, que un paquete de BASIC residente. Sólo cuando comencemos a escribir programas de tamaño significativo necesitaremos recurrir a los procesos más complejos que hemos descrito.

Cada sistema posee su propio juego de instrucciones para controlar el editor y el compilador, y usted debe remitirse a la documentación correspondiente. Con mucha frecuencia, todo cuanto se requiere es una simple E para editar, C para compilar y R para ejecutar. Lo que nos interesa es la sintaxis correcta para entrar un programa, independientemente de lo trivial o complejo que el mismo pueda ser. Por suerte, el PASCAL está tan estandarizado que casi no serán necesarios los muchos "complementos" a la programación que hemos venido ofreciendo para las distintas versiones de BASIC (si bien más avanzado el curso sí tendremos que incluir algunas excepciones menores). Así pues, veamos nuestro primer programa completo en PASCAL:

```
Program Uno (output);
Const
   Mensaje='Programacion en Pascal';
Begin
   write (Mensaje)
End.
```

Antes de estudiar en profundidad este ejemplo, pruebe de entrarlo, compilarlo y ejecutarlo. Si obtiene alguna queja por parte del compilador, lea atentamente el mensaje de error y vea si puede detectar qué es lo que está mal. Todos los símbolos se deben presentar exactamente tal como se indican. Tenga especial cuidado con el punto y coma tras la primera y la tercera líneas, y el punto al final del programa. Cuando haya ejecutado con todo éxito este programa, verá visualizado en la pantalla el siguiente mensaje:

```
Programacion en Pascal
```

El programa es trivial, pero demuestra la forma general que tendrá cada módulo (programa, procedimiento o función). Hay tres partes separadas:
1. El encabezamiento, en este caso un encabezamiento de programa.
2. Declaraciones y definiciones, aquí hay una única definición de constante.
3. El "cuerpo", que contiene todas las sentencias ejecutables.

Las exigencias de sintaxis del PASCAL, al menos para los fundamentos del lenguaje, se pueden definir mejor mediante "diagramas de sintaxis". Éstos son como el mapa de carreteras de un sistema de una sola dirección. La ruta legal a través del diagrama avanza desde arriba a la izquierda hasta abajo a la

Paquete: HISOFT PASCAL
Para: Spectrum, Amstrad, máquinas MSX
Descripción: Una implementación ligeramente no normalizada, pero, no obstante, una ganga. El paquete viene con su propio editor y con una biblioteca de gráficos de tortuga

Paquete: Acorn ISO PASCAL
Para: BBC Modelo B, Electron
Descripción: Una oferta excelente. Hay dos compiladores: uno en ROM, con un editor semiinteligente y compilación de memoria a memoria; el otro es un sistema basado en disco para quienes posean el segundo procesador (6502)

Paquete: Oxford PASCAL
Para: Commodore 64
Descripción: Otra ganga. Se espera que Limbic Systems, la empresa que produce el paquete, saque al mercado versiones para el BBC Micro y el Spectrum en fechas muy próximas





derecha, y todos los casilleros que atravesemos son o bien una "entidad sintáctica" (es decir, que se representa a sí misma) contenida en un casillero de bordes redondos, o bien otro elemento descrito en alguna otra parte mediante un diagrama de sintaxis diferente, indicado por el casillero rectangular que lo contiene.

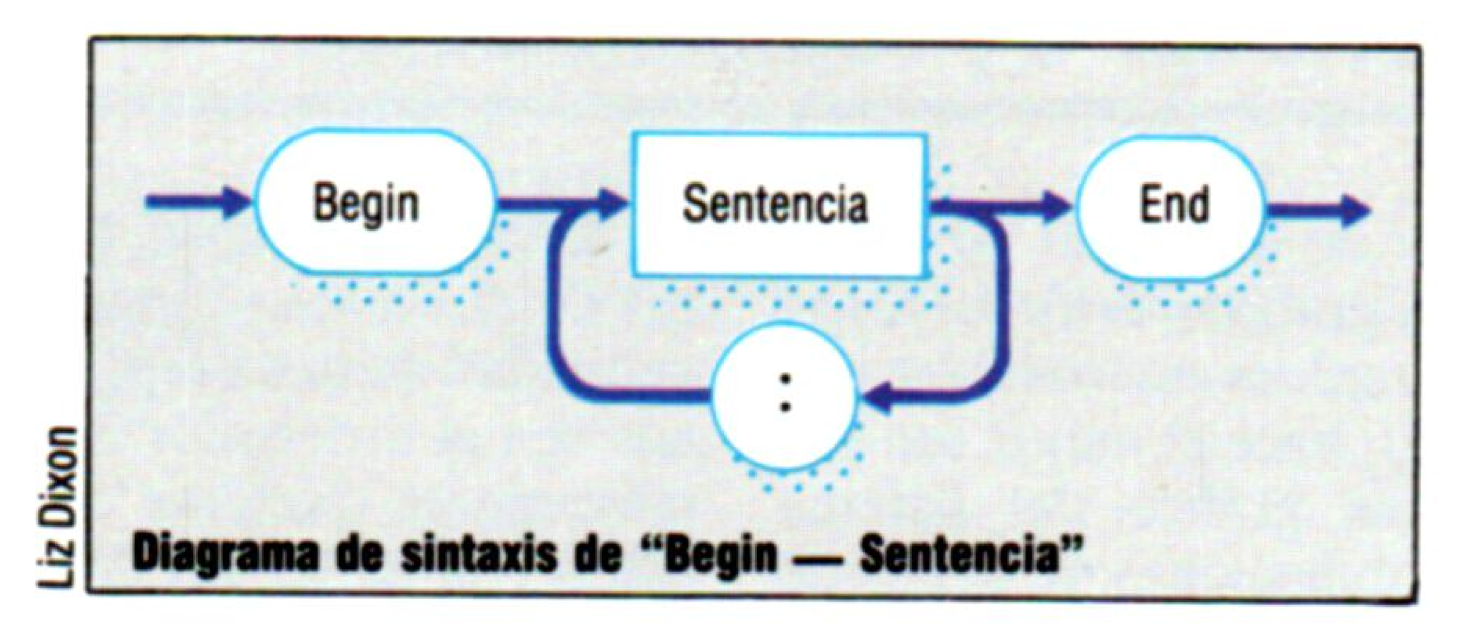


Diagrama de sintaxis de "Begin — Sentencia"

Refiriéndonos primero al diagrama global de un programa, se puede apreciar que las palabras Begin y End están definidas como integrantes del vocabulario del PASCAL, y para elaborar su significado no se requiere ningún otro diagrama. De hecho, el PASCAL reconoce sólo 35 palabras con un significado fijo, cuya lista completa ofrecemos aquí por razones de completitud. Nuestro primer programa sólo hace uso de cuatro de ellas: Program, Const (de *constant*: constante), Begin y End. La palabra que sigue a Program es un "identificador" que identifica el nombre del programa y puede ser cualquier identificador legal que se elija.

Comenzando por una letra y usando sólo letras o dígitos, la cantidad de nombres que podríamos utilizar es inmensa. Sin embargo, como es lógico, es imposible emplear una palabra reservada. P. ej.:

```
Nombre
PASCAL
ProgramUno
N
XYZ123
Direccion12
FloraMacDonald
UnIdentificadorMuyLargoRealmente
```

son todos legales, mientras que:

```
Prog-1
DIEZ%
Trastos.Jack
and
Time$Square
No-lo-se
101Dalmatas
Es igual a
```

son todos ilegales, bien porque poseen símbolos que no son alfanuméricos, bien porque empiezan por un número o (en el caso de and) son alguna de las palabras reservadas del PASCAL. El último ejemplo es ilegal porque se utiliza un espacio para separar palabras, siendo legales las palabras que lo componen (Es, igual y a). En PASCAL, como en castellano o inglés, las letras en minúscula y en mayúscula no suponen ninguna diferencia de significado, si bien algunas versiones no estandarizadas exigen que las palabras reservadas vayan en mayúsculas.

Aparte de los espacios y el final de una línea, en la sintaxis del PASCAL hay otro elemento que se puede emplear como separador: un comentario. Éste puede aparecer en cualquier lugar del texto, excepto, por supuesto, en medio de las palabras. Los comentarios se delimitan mediante "llaves" ({}).

Seamos más audaces y entremos algo que se parezca más a un auténtico programa en PASCAL:

```
Program ProgramDos (input,output);
        {Da el cuadrado de un numero}
Const
        Aviso='Entre un numero:';
Var
               numero :integer;

Begin
   WriteLn;
   WriteLn;
   write (Aviso);
   read (numero);
   WriteLn (numero,'al cuadrado es',numero*numero)
End.
```

Observe que ahora incluimos en el encabezamiento el identificador input. El PASCAL requiere input y output, que identifican los archivos externos con los cuales se comunicará el programa. En un micro, normalmente serán el teclado y la pantalla, respectivamente. Mediante la inclusión de input podemos ahora leer desde el teclado gracias al procedimiento estándar read. Al igual que write, todos los "parámetros" se deben listar entre paréntesis.

La memoria utilizada para almacenar estos parámetros queda reservada por la declaración Var, en este caso para un único número entero. A diferencia del BASIC, que por lo general sólo puede distinguir entre números y series (mediante el empleo de un signo dólar después del identificador), la gama de tipos de datos de que dispone el PASCAL es casi ilimitada. Por consiguiente, es importante informar al compilador sobre cuánta memoria debe reservar para almacenar cada dato. Usted siempre debe declarar las variables simples en una declaración Var.

El cursor permanecerá posicionado inmediatamente después del aviso, igual que si hubiéramos empleado una sentencia PRINT del BASIC con un punto y coma final. Esto es exactamente lo que queríamos al utilizar el procesamiento incorporado write del PASCAL. Siempre que se requiera una nueva línea, debemos emplear el procedimiento alternativo, WriteLn. La Ln es una contracción de la palabra Line (línea), y es útil usar una W y L mayúsculas para indicar el comienzo de cada una de las palabras que lo componen. Una sentencia WriteLn simple sin parámetros creará sólo una nueva línea.

El PASCAL nos permite diferenciar entre datos que varían y datos que permanecen constantes a lo largo de la ejecución del programa. Observe que la definición Const emplea un signo de igualdad, mientras que la declaración Var utiliza dos puntos.

Las 35 palabras del PASCAL

					In	Not	Procedure	Set	Until
And	Case	Do	End	Function	Label	Of	Program	Then	Var
Array	Const	Downto	File	Goto	Mod	Or	Record	To	While
Begin	Div	Else	For	If	Nil	Packed	Repeat	Type	With

Hacia adelante

He aquí otro paquete de análisis del camino crítico, esta vez sobre máquinas como el IBM-PC y compatibles

Aunque el paquete *Project* de Microsoft no posee la clase de facilidades para gráficos de *MacProject* (o de su programa rival, *Graphics environment manager*, de Digital Research), hace el mejor uso posible del limitado potencial gráfico del sistema IBM. Por ejemplo, las combinaciones de signos de igualdad o de guiones con el signo de mayor puede producir algunas útiles flechas:

======>   ------>

La primera se puede visualizar en negrita para indicar los caminos críticos, y la segunda en un tipo más suave para reflejar aquellas actividades que no son críticas. Sin embargo, los factores más sobresalientes que se deben considerar son de una naturaleza más comprometida que ésta. Ellos son:

- ¿Cuál es el grado de flexibilidad del programa, y cuál es su capacidad de respuesta a la clase de cambios que probablemente sean necesarios a medida que se desarrolle el proyecto?
- ¿Con qué claridad visualiza las relaciones existentes entre las actividades, y en qué medida contribuye la misma a la eficaz dirección del proyecto?
- ¿Se puede utilizar el programa como medio auxiliar para la planificación inicial, o los usuarios deben primero desarrollar por completo sus ideas?

Respondiendo primero a la última pregunta, el usuario en realidad necesita clasificar el proyecto sobre el papel antes de digitar la primera actividad. De hecho, el manual afirma de forma inequívoca:

"Antes de entrar actividades en el Microsoft Project, ha de disponer de la siguiente información sobre cada actividad:
En qué consiste (descripción)
Cuánto lleva (duración)
Qué se ha de hacer con anterioridad (predecesores)
Cuándo comienza (fecha de inicio)."

Para poner de relieve la importancia de tal preparación, el manual ofrece un apéndice de diez páginas, *"Defining tasks by a work breakdown structure"* (Definiendo las tareas mediante una estructura de análisis del trabajo), que afirma:

"La dirección exitosa de un proyecto empieza al comienzo de un proyecto. Primero, debe enunciar claramente los objetivos, la finalidad, las limitaciones y las especificaciones del proyecto. Luego debe definir el conjunto de actividades específicas que constituyen el proyecto.

"Ningún programa para ordenador puede compensar las actividades definidas de forma inadecuada o sin precisión. La definición de las actividades puede tener implicaciones en la planificación de los recursos, su asignación y su control, así como en el control de costos y presupuesto. La definición inexacta de las actividades probablemente ocasione graves problemas una vez iniciado el proyecto."

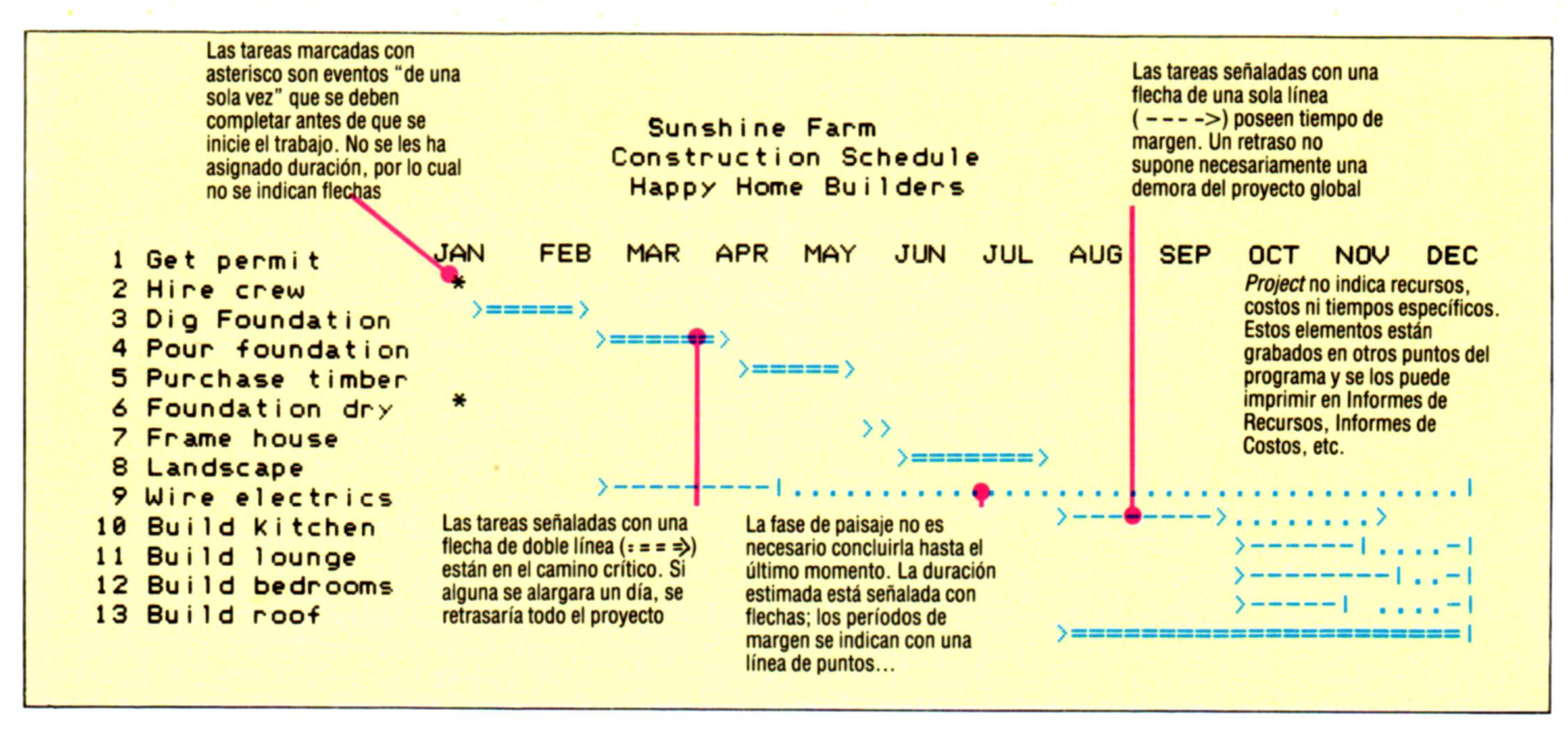
Es difícil que quienes posean experiencia en técnicas de dirección se muestren en desacuerdo con la necesidad de una preparación adecuada. Pero quienes ven al ordenador como una herramienta cualitativamente diferente de todas las otras, actuando como una extensión de la mente del usuario y no como una extensión de las yemas de sus dedos, quizá piensen que la tarea del software es, precisamente, contribuir a evitar actividades definidas inadecuada o inexactamente. Pero, por cierto, Microsoft Project ayuda al usuario a detectar las prioridades. Incluyendo el recurso inicial del lápiz y el papel, también ofrece al usuario muchísimas oportunidades para introducir cambios.

Cuando se conecta por primera vez el ordenador, la fecha y la hora, que se incluyen como parte de secuencia de entrada *(sign-on)*, determinarán el calendario empleado por el programa; pero el calendario se puede cambiar con bastante facilidad, incluso tras haber digitado varias actividades. De alterarse la hora asignada a alguna actividad, tan pronto se entra la modificación, se vuelve a calcular el diagrama del camino, cambiando algunas flechas de actividades a negritas ===> (críticas), otras a ligeras -----> (no críticas). Ello no sólo hace que observar la relación entre actividades sea muy fácil, sino que también destaca de forma instantánea un cambio que, de lo contrario, no le habría resultado obvio al usuario. Por todo ello, parece que Microsoft ha aplicado la experiencia que obtuvo con la producción del *Multiplan*, una de las hojas electrónicas de mayor éxito, después de que en 1979 el pionero *Visicalc* inaugurara el género. En realidad, usar el *Microsoft Project* es casi como utilizar una hoja electrónica, y quienes estén familiarizados con el empleo de las mismas se sentirán casi de inmediato a sus anchas con *Project*.

Tras una breve visualización del logotipo de Microsoft, se visualiza la pantalla de actividades (vacía, si no se hubiera especificado ningún nombre de archivo), con números del 1 al 19 a la izquierda de cada fila y la escala de tiempo a lo largo de la parte superior, empezando por la fecha de hoy, si se ha entrado inicialmente. Ésta se puede modificar mediante la entrada OPTIONS del menú de la parte inferior de la pantalla (nuevamente al estilo de *Multiplan*), que incluye asimismo:

```
BLANK CALENDAR DELETE EDIT GOTO HELP INSERT
MOVE PRINT QUIT RESOURCE SORT TRANSFER
XTERNAL
```

Debajo de la frase *"Select option or type command letter"* (Seleccione opción y pulse letra de comando), se visualiza la opción en uso en ese momento (inicialmente ACTIVITY) y, en el extremo inferior derecho, el nombre del *Project* en uso, al cual de forma automática se le da el título TEMP hasta que el usuario le asigne algún otro.



Proyecto de construcción
Este diagrama representa un plan para la construcción de una casa. Las fechas aparecen a través de la parte superior del diagrama, y cada una de las tareas aparece, por orden de operación, a lo largo del margen izquierdo. La duración de cada tarea se representa mediante flechas construidas con caracteres del teclado (<>.=−)

La escala de tiempo se puede regular en días, semanas o meses y se la puede cambiar en cualquier momento para contribuir a ofrecer una visión general de todo el proyecto a simple vista. Con una escala de tiempo diaria sólo se pueden ver, supongamos, las fechas entre el 1 de enero y 2 de marzo, mientras que la escala de tiempo mensual permite ver a simple vista dos años y tres meses. Las teclas derecha e izquierda para control del cursor se emplean para desplazar hacia los lados la pantalla ACTIVITY, al estilo habitual de una hoja electrónica. Mediante la opción CALENDAR, el usuario programa primero las vacaciones y otros días no laborables; los sábados y domingos normalmente se planifican como días festivos. Ello, sin embargo, se puede alterar, pero se debe realizar para cada día individual. Los meses posteriores se pueden visualizar mediante la tecla <PAGE DOWN>, y los meses anteriores se pueden recuperar con <PAGE UP>.

Al igual que sucede con la mayoría de las opciones de fecha en el IBM y máquinas compatibles, debe tenerse cuidado en utilizar el orden norteamericano de MM/DD/AA (p. ej., 01/07/85 es 7 de enero de 1985, y no 1 de julio de 1985). Una vez ajustado el calendario, puede ser guardado (SAVE) utilizando la opción TRANSFER en el estilo *Multiplan* habitual. El programa añade el sufijo .CAL para distinguirlo de los archivos de actividades (sufijo .ACT) y de recursos (.RES).

El usuario comienza luego a digitar actividades, el tiempo que lleva cada una, y anunciando las actividades previas de las que depende (sus "predecesoras"), con las dependencias múltiples separadas mediante comas. La columna de Activities Titles (nombres de las actividades) está limitada a 15 caracteres y no se puede ensanchar. Sin embargo, de digitarse más de 15 caracteres, éstos se retendrán en la memoria y se incluirán en las salidas impresas. A medida que se va digitando cada actividad (con su duración, su(s) predecesora(s) y su fecha de inicio), se van visualizando inmediatamente las relaciones entre las diversas actividades, con el tiempo de margen (período durante el cual se puede retrasar una actividad sin que se atrase el proyecto total) representado en forma de puntos. En este momento se asignan, asimismo, los recursos. Apenas se entra un recurso, se puede llamar a la Resource Table (tabla de recursos) para controlar los resultados.

Una vez entrados los datos, se puede imprimir el plan completo en una base de página a página. Pero si se ha instalado una ficha para gráficos y conectado la impresora apropiada (impresora para gráficos del IBM-PC o las Epson MX-80, MX-100, FX-80 o FX-100), se puede volcar todo el proyecto de forma apaisada y obtener una impresión a lo largo de la longitud del papel. Esto es útil, dado que significa que se puede imprimir un plan completo sin que sea necesario cortar y pegar las salidas impresas.

Asimismo, es posible obtener informes detallados acerca de cada una de las actividades, reflejando las fechas de inicio y terminación más tempranas y más tardías, los recursos, las predecesoras y los tiempos de margen, y una tabla de actividades para el proyecto en conjunto visualizando la misma información pero de forma menos detallada.

En la práctica, la carencia del *Microsoft Project* de gráficos sofisticados representa una desventaja menos importante de lo que se podría pensar en un primer momento, y aunque el manual (excelente) hace hincapié en la necesidad de una preplanificación adecuada, aun así se puede ajustar la información después del segundo o tercer cambio de idea o tras demoras inesperadas. El programa será particularmente amable para quien tenga experiencia con hojas electrónicas, en especial el *Multiplan*, porque su método de operación es muy similar. En operación el *Microsoft Project* es más rápido que el *MacProject* de Macintosh, pero su precio es casi tres veces superior.

Microsoft Project: Para máquinas MS-DOS
Distribuido por: Microsoft Ltd, Piper House, Hatch Lane, Windsor, Berkshire, Gran Bretaña
Autores: MAS, Seattle, Estados Unidos
Formato: Disco

Se reanudan los juicios

Nuestra comentarista Stephanie Brittain hace una valoración crítica de dos recientes programas de juegos escritos para el Commodore 64



Percy the potty pigeon

(Percy, la paloma chiflada)
Gremlin Graphics

Conducir en Londres entraña sus peligros, incluyendo los traumas que puede causar la conocida aversión de las palomas a volar por donde muy bien podrían ir andando.

Percy, la paloma de este juego, está completamente chiflada. Para ella no hay nada mejor en el mundo que volar por sobre la línea del horizonte de la gran ciudad arrojando bombas a los coches que circulan por la autopista: una sensata narración de carnicería vial. Si se la controla mediante la palanca de mando, Percy puede que no sea destructiva del todo: le agrada abalanzarse sobre la autopista (evitando a los vengativos motoristas) para recoger ramitas, que luego lleva hasta el nido más cercano de los árboles que hay junto al borde del camino. En los niveles de juego superiores, la mano que maneja la palanca de mando debe evitar a los gorriones que roban las ramitas y a los halcones que roban la paloma.

Ciertamente, *Percy the potty pigeon* no es un programa por el que Gremlin Graphics pueda sentirse demasiado satisfecha.



Ian McKinnell

Caesar the cat

(El gato Caesar)
Mirrorsoft

Caesar es un pequeño y astuto gatito negriblanco cuyo objetivo consiste en limpiar de ratones la "despensa".

La despensa en cuestión en realidad es una pantalla repleta de alimentos y cacharros por la cual campan a sus anchas los ratones, comiéndose la comida hasta que Caesar los atrapa.

Se pierden puntos si Caesar se estrella contra una "pared" lateral de la pantalla, choca contra algún cacharro o tarda demasiado tiempo en capturar su presa.

Una vez que consigas dominar el primer nivel, puedes perseguir ratones más veloces en los dos niveles superiores: y es aquí donde comienza realmente la diversión. Ciertamente, Mirrorsoft ha acertado de lleno con esta aventura de felino y roedores. No existe ningún juego comparable a éste en cuanto a originalidad. ¡Les gustará hasta a los amantes de los perros!

En resumen, *Caesar the cat* constituye un auténtico acierto por parte de los programadores de Mirrorsoft.



Portátil avanzado

ACT ha utilizado la más reciente tecnología para darle al Apricot Portable un aire futurista

La máquina portátil representa una de las áreas de mayor crecimiento de la microinformática. Se pretende que el hombre de negocios pueda desplazarse con su micro por el mundo y trabajar allí donde haya una adecuada fuente de alimentación eléctrica. En la práctica, no siempre es así como se utiliza la máquina. A menudo, el ordenador portátil se conserva en casa y el trabajo realizado con una máquina compatible en la oficina se lleva a casa por la tarde para acabarlo.

ACT (Apricot) es uno de los más recientes entre una larga serie de fabricantes de ordenadores que ha producido una máquina portátil que es compatible con su muy vendida línea de ordenadores de sobremesa.

El Apricot Portable se suministra en una sólida carcasa plástica cuyo tamaño y forma son el de un maletín, pero con un saliente en el fondo para dar cabida a la pantalla. La carcasa se abre presionando dos cierres a los lados del asa, aunque en la máquina que examinamos ello resultó bastante dificultoso.

Análisis del hardware

En el interior, el teclado y el ordenador se sostienen a cada lado de la carcasa mediante tiras. Con la caja también se proporcionan dos manuales, un micrófono y un ratón/bola de mando. Lo curioso es que no se suministra ningún cable para unir el teclado y el ratón al ordenador. Ello se debe a que los datos se transmiten a través de un haz infrarrojo. La información se transmite y se recibe mediante dos pequeños focos que hay en el borde trasero del teclado y en la parte delantera del ratón. Las señales emitidas por estos dispositivos las recoge un dispositivo receptor que hay en el ordenador. De este modo, el usuario puede trabajar con el teclado sobre el regazo y tener la máquina sobre una mesa a algunos metros de distancia.

Este sistema infrarrojo posee dos desventajas. En primer lugar, si en una misma habitación se estuviera operando con varios Apricot Portable, las señales infrarrojas de las diferentes máquinas podrían interferirse mutuamente. Para evitar esto, Apricot ha proporcionado un tubo ligero. Éste es un cable de fibra óptica que conecta el teclado con el ordenador de la forma convencional. El segundo problema aparece cuando se utiliza el ratón. Si el usuario coloca el teclado directamente frente al ordenador, con frecuencia bloqueará las señales provenientes del ratón. Ello se debe a que el foco receptor de la máquina queda casi al ras de la superficie de la mesa. Por otra parte, si corre el teclado hacia un lado, puede que el ordenador no sea capaz de captar sus señales.

Uno de los problemas de producir una versión portátil de una máquina de sobremesa es comprimir todas las funciones en un espacio más reducido. El teclado plantea dificultades especiales y los fabricantes se han valido de toda clase de técnicas ingeniosas para empaquetar todas las funciones necesarias en un teclado más pequeño y más ligero. Al traducir el teclado Apricot a la versión portátil, el fabricante ha prescindido de la visualización de calculadora LCD y ha colocado las 87 teclas una bien junto a la otra. El trazado es idéntico al del Apricot de sobremesa, a excepción de las teclas de función, que están situadas en el extremo derecho en vez de arriba del trazado principal.

En parte para compensar la pérdida de la visualización LCD, que en la versión de sobremesa también se utiliza para intrucciones que requieren una única pulsación de tecla, Apricot ha añadido una tecla de función extra, lo que hace un total de nueve, y una tecla para visualizar la hora y la fecha. Las teclas QWERTY del lado izquierdo del teclado están flanqueadas por teclas de control, estando las teclas del cursor abajo a la derecha. Entre las teclas de máquina de escribir y las teclas de función hay un teclado de calculadora, con las teclas numéricas en una tonalidad de gris más clara que la del resto del teclado.

Si bien el aspecto del teclado es sumamente elegante, quizá las teclas estén situadas demasiado cerca las unas de las otras como para resultar cómodas. En vez de las convencionales teclas indentadas, el Portable posee un teclado nivelado estilo QL. Ello hace que el teclado sea más compacto, dado que las teclas no sobresalen del cuerpo principal, pero a muchos mecanógrafos usar este diseño

Tecnología de vanguardia
El Apricot Portable utiliza la tecnología más reciente. La máquina está equipada con una pantalla para textos de 80 por 25, que permite visualizar una amplia variedad de aplicaciones MS-DOS. En la fotografía también vemos el teclado y el ratón/mando de bola: éstos no están conectados al ordenador en el sentido convencional, sino que transmiten señales a través de un haz infrarrojo. En el lado derecho está el micrófono, que le proporciona a la máquina la facilidad de reconocer la voz



Chris Stevens

de teclado les resultará difícil, puesto que a los dedos les es difícil diferenciar las teclas. Lo compacto de esta disposición pone de relieve este problema: no hay ningún vacío entre las teclas de máquina de escribir, las de calculadora y las de función. No obstante, el trazado posee sus compensaciones. En razón del diseño, Shift, Stop y Caps Lock son el doble de grandes que las del Apricot, lo que las hace más fáciles de encontrar.

Encima del teclado y en el lugar de la visualización de calculadora hay cuatro botones para llevar a cabo funciones extras. Hay un botón de Reset, que carga de nuevo el disco que esté en ese momento en la unidad de disco. El segundo botón le permite al usuario modificar la velocidad de repetición de las teclas. El tercero llama a una rutina que pone a cero la hora retenida en el ordenador, y el cuarto traba el teclado. Este último es útil cuando se está utilizando el ratón o la entrada por habla y no desea que la pulsación accidental de una tecla produzca la entrada de información falsa.

La pantalla LCD incorporada está instalada en el cuerpo principal del ordenador. Su nitidez es más pobre que la de máquinas comparables, aun cuando el claro fondo verde oliva sea más brillante que la mayoría. A la derecha de la pantalla hay un micrófono, que se conecta mediante un delgado cable que se enchufa en un conector microjack situado a la izquierda del ordenador.

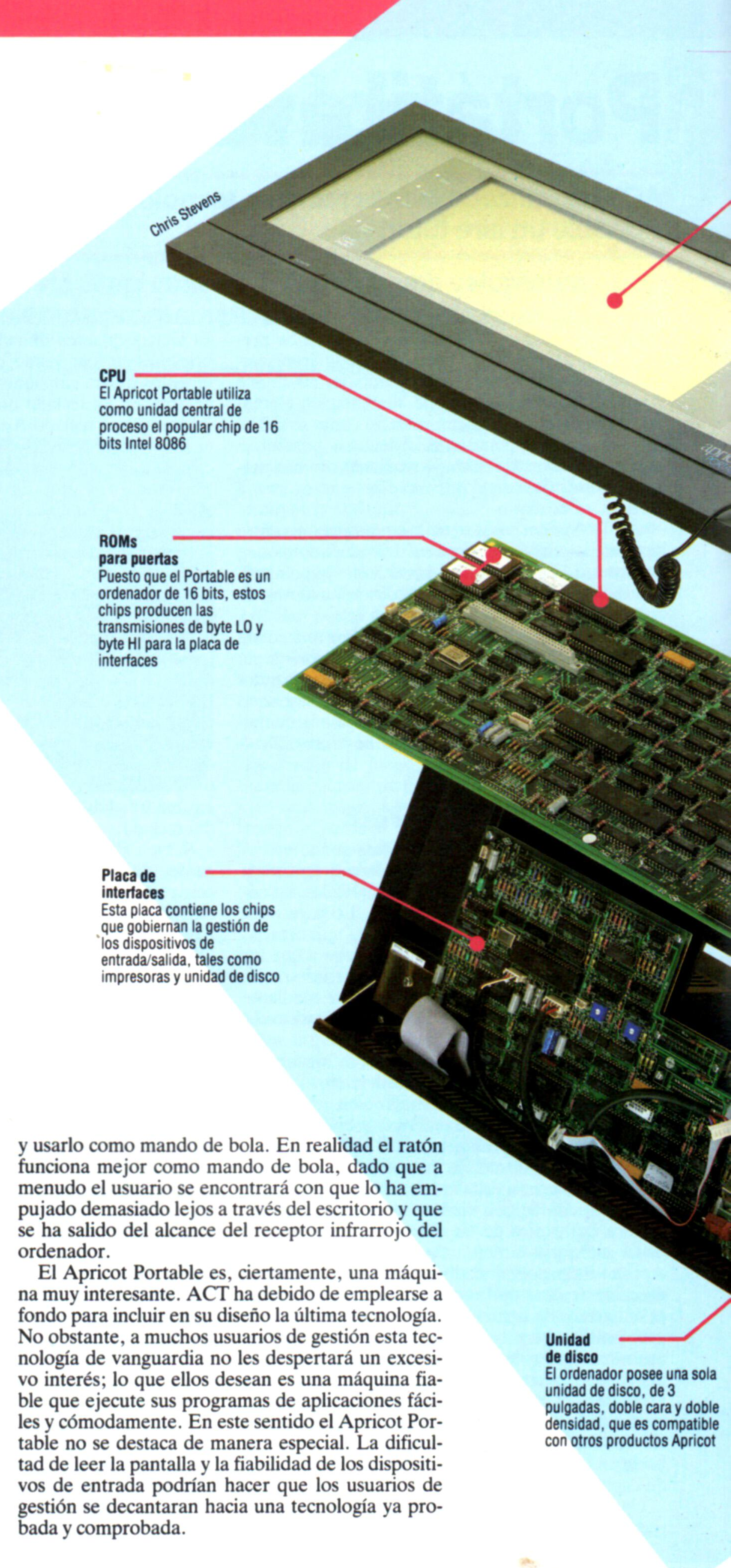
Reconocimiento de voz

Quizá la característica más interesante del Apricot Portable sea su programa para reconocimiento de voz, que reconoce instrucciones orales, las pasa a la entrada y después las ejecuta de la forma normal. La modalidad Voice Driven Application (aplicación activada mediante la voz) acepta vocabularios de hasta 63 palabras para cualquier programa de aplicaciones, como el *WordStar* o el *SuperCalc*. Un programa de entrenamiento de voz le preguntará al usuario si la voz a utilizar será masculina, femenina o infantil, y si habrá o no mucho ruido de fondo. Después de pronunciar varias veces una instrucción, el ordenador aceptará y comparará diversas variaciones. Sin embargo, aun así malinterpretará palabras y con frecuencia también las ignorará. Por este motivo se recomienda no emplear de forma oral DELETE y FORMAT.

Al igual que el resto de la gama Apricot, el Portable utiliza discos Sony de 3 pulgadas para retener los programas de aplicaciones. La única unidad de disco está situada en el lado derecho del ordenador. En la parte posterior de la máquina hay tres interfaces cubiertas por una carcasa plástica protectora. Éstas son una ranura en paralelo Centronics para impresoras, un conector RS232 para un modem externo u otro dispositivo en serie y una puerta para palanca de mando tipo Atari. Esta ranura para palanca de mando no se provee para juegos sino para instalar un ratón o un mando de bola en aquellas circunstancias en las que el ratón infrarrojo no resulte adecuado.

El ratón infrarrojo activado a pilas que se suministra con el Portable está diseñado de forma inteligente, de modo que pueda ser utilizado bien como un ratón que se pueda deslizar por la superficie de una mesa o escritorio, o bien, invertido, para sostenerlo en la mano o mantenerlo en una posición fija y usarlo como mando de bola. En realidad el ratón funciona mejor como mando de bola, dado que a menudo el usuario se encontrará con que lo ha empujado demasiado lejos a través del escritorio y que se ha salido del alcance del receptor infrarrojo del ordenador.

El Apricot Portable es, ciertamente, una máquina muy interesante. ACT ha debido de emplearse a fondo para incluir en su diseño la última tecnología. No obstante, a muchos usuarios de gestión esta tecnología de vanguardia no les despertará un excesivo interés; lo que ellos desean es una máquina fiable que ejecute sus programas de aplicaciones fáciles y cómodamente. En este sentido el Apricot Portable no se destaca de manera especial. La dificultad de leer la pantalla y la fiabilidad de los dispositivos de entrada podrían hacer que los usuarios de gestión se decantaran hacia una tecnología ya probada y comprobada.

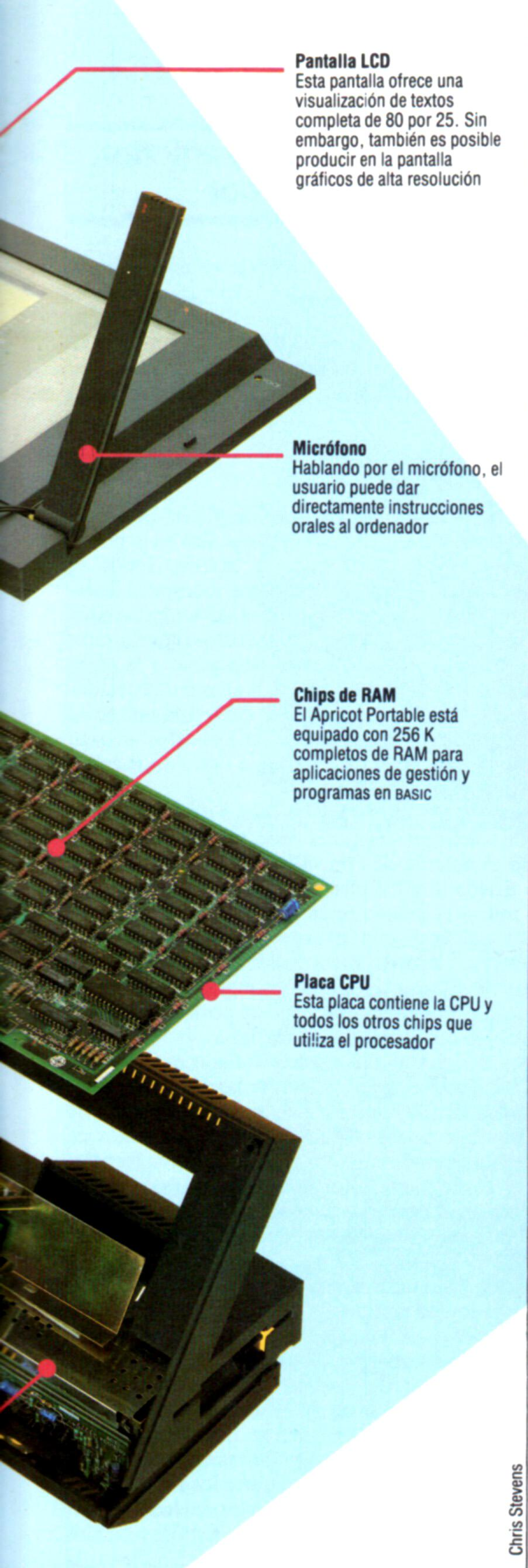

Chris Stevens

CPU
El Apricot Portable utiliza como unidad central de proceso el popular chip de 16 bits Intel 8086

ROMs para puertas
Puesto que el Portable es un ordenador de 16 bits, estos chips producen las transmisiones de byte LO y byte HI para la placa de interfaces

Placa de interfaces
Esta placa contiene los chips que gobiernan la gestión de los dispositivos de entrada/salida, tales como impresoras y unidad de disco

Unidad de disco
El ordenador posee una sola unidad de disco, de 3 pulgadas, doble cara y doble densidad, que es compatible con otros productos Apricot

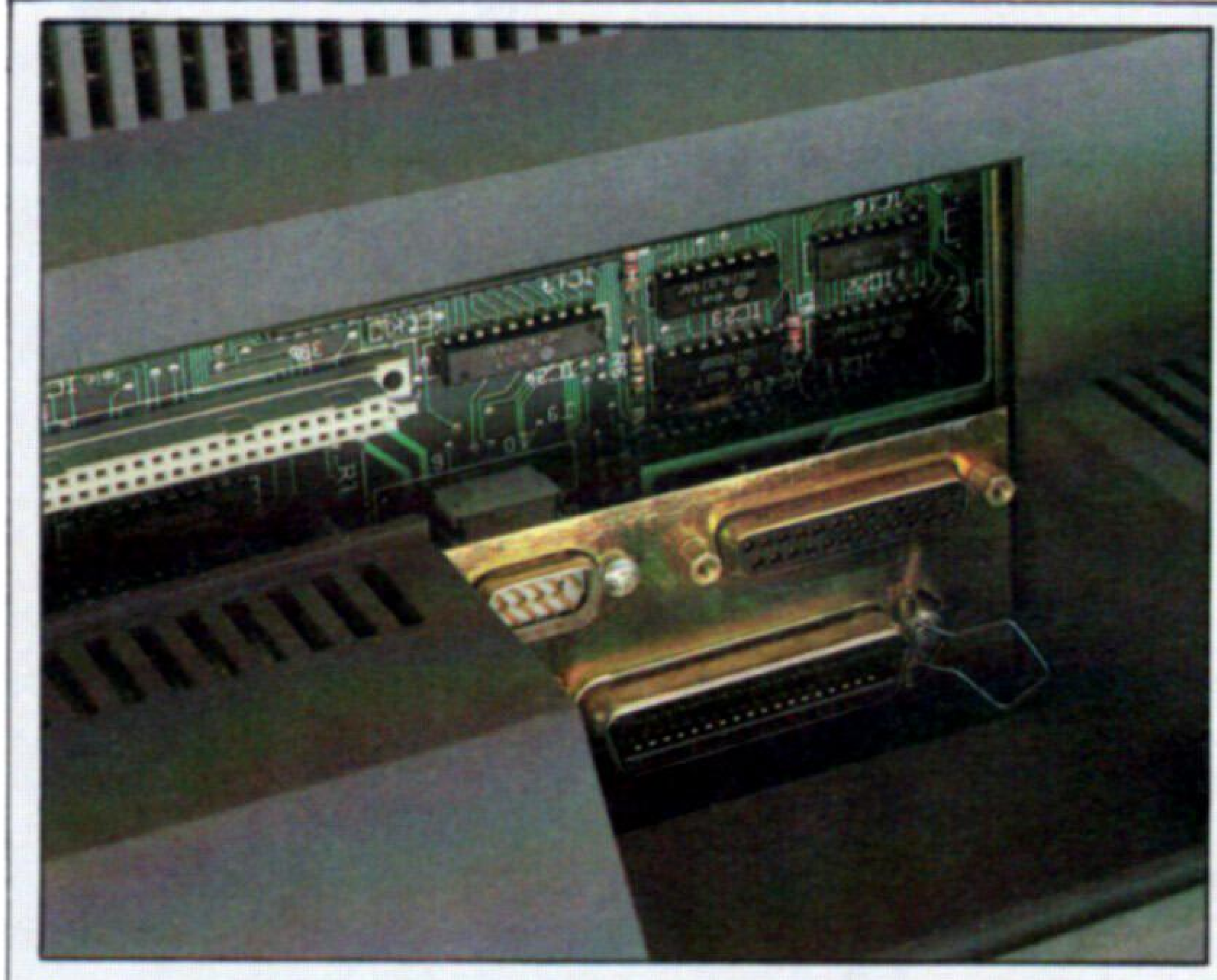


Pantalla LCD
Esta pantalla ofrece una visualización de textos completa de 80 por 25. Sin embargo, también es posible producir en la pantalla gráficos de alta resolución

Micrófono
Hablando por el micrófono, el usuario puede dar directamente instrucciones orales al ordenador

Chips de RAM
El Apricot Portable está equipado con 256 K completos de RAM para aplicaciones de gestión y programas en BASIC

Placa CPU
Esta placa contiene la CPU y todos los otros chips que utiliza el procesador



Interfaces para periféricos
En la parte posterior del ordenador, protegidas bajo una carcasa plástica, están las interfaces para periféricos. Abajo está la puerta en paralelo Centronics y, encima de ella, a la derecha, el conector RS232 (que permite conectar un modem externo). A la izquierda hay una puerta a través de la cual se puede conectar un ratón o un mando de bola convencionales



Mando de bola infrarrojo
Haciendo girar la bola, la flecha del cursor se desplaza en la dirección correspondiente a través de la pantalla. Cuando el cursor está señalando el icono elegido, pulsando uno de los botones del costado se accede a la aplicación escogida. El dispositivo se puede dar vuelta, de modo que la bola ruede, lo que le permite al dispositivo actuar a modo de ratón



Chris Stevens

Software Apricot
Con el ordenador se suministran algunos paquetes de aplicaciones que permiten ejecutar varios programas diferentes. Estas aplicaciones están retenidas en discos Sony de 3 pulgadas. Entre los paquetes están el *SuperCalc* (hoja electrónica), el *SuperPlanner* (fichero de tarjetas y calendario) y el *SuperWriter* (procesador de textos)

APRICOT PORTABLE

DIMENSIONES
450 × 335 × 200 mm

CPU
Intel 8086

MEMORIA
32 K de ROM y 256 K de RAM, de los cuales hay 211 K libres para programas de aplicaciones

PANTALLA
Pantalla LCD electrofluorescente con una resolución de texto de 80×25 caracteres y una resolución para gráficos de 640×256 pixels. Si se enchufa el ordenador a una pantalla externa, se pueden visualizar hasta 16 colores

INTERFACES
En serie RS232, en paralelo Centronics y una conexión para ratón. También hay una conexión de "tubo de luz" para utilizar el teclado basado en infrarrojos y un conector hembra para el micrófono

UNIDADES DE DISCO
El Portable contiene una unidad para discos Sony de 3 pulgadas, doble densidad y 720 K

SISTEMAS OPERATIVOS
MS-DOS, CP/M-86 y Concurrent CP/M

TECLADO
Teclado compatible con IBM, conteniendo 87 teclas con diez teclas de función. Hay, asimismo, cuatro botones de función

DOCUMENTACION
Un manual Starter Pack, que ofrece detalles sobre cómo instalar el ordenador y cómo utilizar el sistema operativo y las facilidades para reconocimiento de voz. El manual Applications Pack ofrece una guía para las aplicaciones de *SuperCalc*, *SuperPlanner* y *SuperWriter*

VENTAJAS
El Apricot Portable es una de las máquinas más avanzadas que existen en la actualidad

DESVENTAJAS
Debido a que la tecnología utilizada es tan avanzada, aún no es infalible. Leer la pantalla LCD puede resultar muy difícil

Llenando el vacío

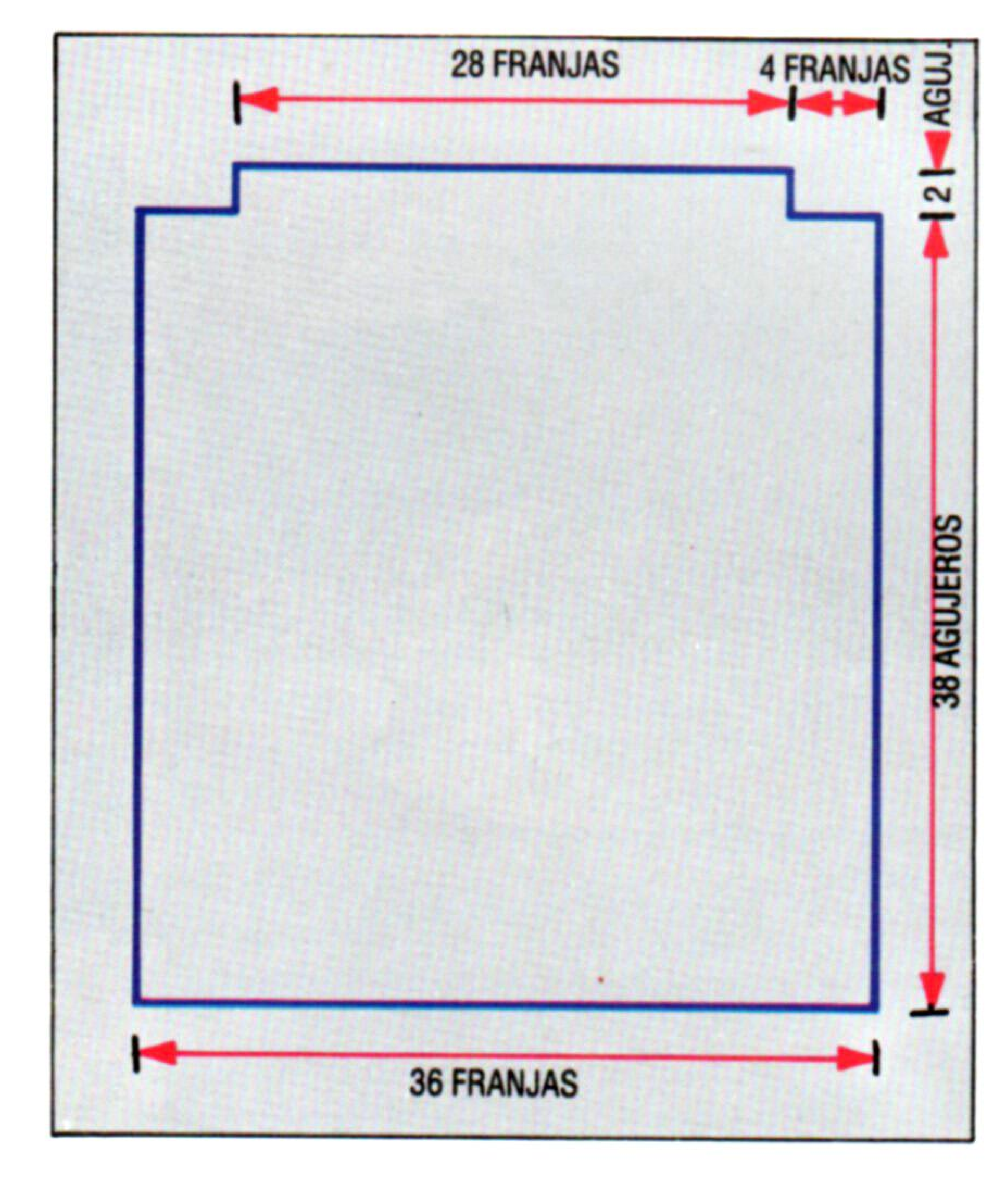
Ahora construiremos la interface diseñada en el capítulo anterior, que le permitirá al usuario del Spectrum controlar el robot

El conector marginal diseñado para el Spectrum habrá de tener anulada la quinta posición desde la izquierda (mirando al conector). Por consiguiente, nuestra primera tarea consiste en insertar un enchufe anulador en el conector marginal; debe enchufarse en la puerta para ampliación. Este enchufe asegura que la interface terminada sólo se pueda insertar en el conector si está orientada correctamente. El enchufe es fácil de construir.

Después de cortar el veroboard de la forma indicada, se puede usar como enchufe anulador una de las esquinas. Primero, pele la pista de cobre y fije el cuadrado cortado de la placa por el borde en la quinta posición del conector marginal (utilizando un poco de supercola), luego corte ambas conexiones de la pantalla. Después efectúe en el veroboard profundos cortes de pistas, usando ya sea una cuchilla o bien una broca espiral de 5 mm. Asegúrese de que cada incisión corte toda la anchura de la pista, ya que, de lo contrario, además de que su interface no funcionaría, incluso se podría dañar el ordenador.

Lista de componentes

Cantidad	Artículo
1	74LS27
1	74LS75
1	74LS126
4	15 K-ohmios, 0,3W
	Carrete alambre estañado de 20 swg
2	Conector DIL de 14 vías
1	Conector DIL de 16 vías
5 m	Cable plano de 20 vías
1	Conector marginal de 2×28 vías y 0,1 pulgadas
1	Conector D de 15 vías
1	Tapa para conector D de 15 vías
1	Veroboard de 36 franjas × 50 aguj.



Observaciones cortantes
Los componentes de la interface para el Spectrum se montan en un trozo de veroboard cortado en la forma que vemos en la ilustración. Use una de las pequeñas esquinas cortadas como enchufe anulador para el conector de la puerta para ampliación. El borde extra en uno de los extremos de la placa aguantará este conector cuando el mismo se suelde en su sitio

Ensamblaje

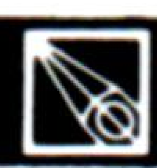
Habiendo efectuado los cortes de pistas, podemos comenzar a ensamblar los componentes en la placa, empezando por los cables de enlace. Éstos se deben cortar holgadamente largos y después insertar con flojedad en la placa. Suelde un extremo y tire del otro hasta tensarlo antes de soldarlo también. Por último, debe cortar bien junto a la placa los trozos de cable sobrantes con un par de cortadores de cable. Cuando todos los cables de enlace se hayan soldado en su sitio, coloque el cable volante aislado, empleando un trocito de uno de los cabos del cable plano.

Después de fijar todos los cables de enlace ya se puede soldar en su sitio el conector marginal. Pero como el sistema de conector marginal en realidad está diseñado para placas de doble cara (como el que utiliza el propio Spectrum) y no para el veroboard, que es de una sola cara, se trata de una operación que tiene su truco. Primero, incline una fila de patillas sobre el conector marginal, como se muestra. Cuando el conector esté posicionado en el tablero, éstas formarán la fila inferior de patillas. Ahora suelde 3 cm de cable estañado en cada una de las de la fila superior. Inserte las patillas inclinadas en la fila de agujeros más próxima al borde de la placa e incline en 90° cada uno de los trozos de cable; insértelos en la otra fila de agujeros (señalados en el diagrama). Luego coloque un poco de supercola en el borde de la placa entre las dos esquinas cortadas y presione el conector marginal en su sitio.

Suelde cuidadosamente cada patilla a las pistas de la placa de franjas. Antes de seguir adelante, compruebe el lado de cobre de la placa y quite, con un cuchillo puntiagudo, cualquier soldadura que pueda estar haciendo puente entre las pistas.

Ahora suelde el resto de los componentes en sus lugares en la placa. Primero se han de soldar los tres conectores DIL y luego las cuatro resistencias de 4 K-ohmios. Por último, inserte los tres chips en su sitio, asegurándose de colocarlos del lado correcto. Las muescas de uno de los extremos de los chips deben apuntar todas en la misma dirección: hacia la derecha cuando se sostiene la placa, con el conector marginal arriba de todo.

Ahora la interface ya está completa. No obstante, antes de enchufarla en el ordenador, es importante comprobarla concienzudamente. Dado que la puerta para ampliación del Spectrum nos otorga ac-



Trazado del tablero

Suelde primero los cables de enlace y resistencias en su sitio en el lado para componentes de la placa, después instale el conector de la puerta para ampliación y los conectores DIL. Asegúrese de que los tres chips estén en la orientación correcta (las muescas deben apuntar hacia el mismo lado). Antes de enchufar a su Spectrum la placa de interface, compare cuidadosamente su trabajo con el diagrama.
En la cara posterior de cobre, asegúrese de que todos los cortes de pistas interrumpan correctamente las pistas, utilizando un medidor para probar la posible continuidad. Los enlaces señalados en verde son conexiones temporales y se deben eliminar una vez realizada con éxito la comprobación

ceso ilimitado a los buses de direcciones y de datos, a las patillas del procesador y a las fuentes de 12 V y 9 V, es importante que, sin habernos percatado de ello, no hayamos conectado estas fuentes de voltaje a la delicada electrónica de los niveles TTL. Primero se debe efectuar una comprobación visual. Repase cuidadosamente toda la placa. Compruebe que no haya puentes de soldadura entre pistas en ningún punto, que todos los componentes y los enlaces de cables estén en el lugar que indican los diagramas y que los chips estén colocados por el lado correcto.

En el próximo capítulo tenderemos el cable plano que conecta el robot a la placa de la interface y veremos versiones para el Spectrum de algunos de los programas que hemos escrito previamente.

CABLES DE ENLACE DE 3 CM
CONECTOR DE PUERTA AMPLIACIÓN
VEROBOARD
COLA
Kevin Jones

Vista lateral
Para poder utilizar un conector marginal estándar para la placa de interface, hemos de efectuar ligeras modificaciones en las patillas. Habiendo eliminado las patillas de la quinta posición, incline el grupo de patillas inferior en ángulo recto y suelde 3 cm de cable estañado al grupo superior. Empuje las patillas a través de los agujeros correspondientes de la placa, comprobando que queden situadas correctamente. La fila inferior de patillas debe situarse en la tercera fila de agujeros y la fila superior, en la sexta. Pegue el conector en su sitio antes de soldar las patillas a la placa

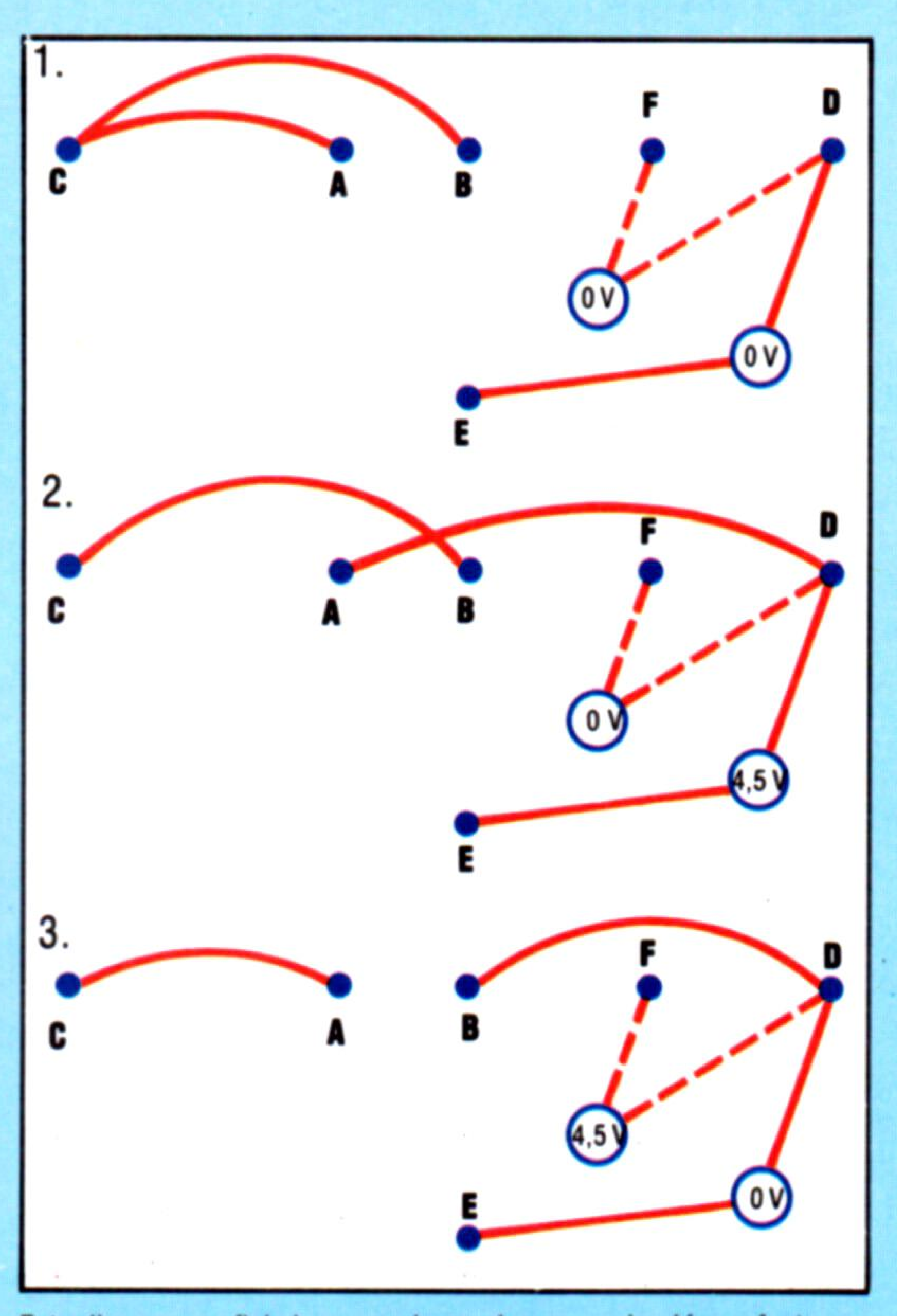
¡ATENCIÓN!

La puerta para ampliación del Spectrum nos permite acceder a la delicada electrónica interna. En la puerta también hay presentes salidas de voltaje que si, inadvertidamente, se dirigieran al bus de datos o al de direcciones, dañarían su ordenador. Compruebe TODAS las conexiones antes de enchufar la interface en la puerta y encenderla

Comprobación de la placa

Antes de enchufar la placa de interface en la puerta para ampliación del Spectrum, podemos realizar una serie de comprobaciones eléctricas para asegurar que la lógica decodificadora esté funcionando correctamente. Para hacerlo necesitamos un tester, una pila de 4,5 V y varios trozos cortos de cable forrado. Primero, realice las dos conexiones que se indican en color verde en el lado de cobre de la placa de circuitos y conecte los terminales de la pila a los puntos adecuados de la placa. Utilice dos trozos de cable forrado para conectar el punto A al punto C y el punto B al punto C. Verifique su trabajo y elimine el exceso de soldadura que pudiera haber entre pistas. Ahora ya podemos comprobar los niveles de voltaje en los puntos E y F. Con la sonda negativa del medidor colocada en el punto D, toque con la sonda positiva primero el punto E y luego el punto F. Estas dos lecturas deben ser de entre 1 V y 0 V. Desuelde el cable entre A y C en C y reconecte el extremo suelto al punto D. Con la sonda negativa de su medidor aún en el punto D, tome nuevas lecturas de E y F. Ahora E debe dar una lectura de 4,5 V y F debe seguir siendo de 0 V. Desconecte el cable de A a D en D, y el cable de B a C en C. Intercambie los extremos sueltos y vuelva a efectuar las conexiones de modo que B se conecte con D y A se conecte con C. Vuelva a tomar lecturas de E y F. Ahora F debe dar 4,5 V y E debe dar 0 V. Si estas pruebas producen las lecturas correctas en los puntos E y F, quite de la placa las conexiones de la pila, los dos cables de enlace verdes y los cables volantes entre A y C, y B y D. Sin embargo, si alguna de estas lecturas no fuera la que indicamos, vuelva a revisar la construcción de toda la placa. La placa de interface no se debe enchufar en el Spectrum bajo ninguna circunstancia hasta que las lecturas del medidor no sean las correctas. En la tabla ofrecemos las diversas conexiones de prueba y las lecturas correctas.

Conexiones	Lecturas
A-C, B-C	E=0 V F=0 V
A-D, B-C	E=4,5 V F=0 V
A-C, B-D	E=0 V F=4,5 V



Este diagrama refleja las conexiones de comprobación a efectuar entre los puntos A, B, C y D, y las lecturas correctas que se deben obtener en los puntos E y F en cada una de las tres etapas de prueba

Fichas archivadas

Consideramos ahora la llamada OSFIND y algunas otras rutinas encargadas de la escritura y recuperación de los datos en su almacenamiento

La OSFIND es llamada en la dirección &FFCE y sirve para abrir un fichero para lo que se llama *acceso al byte*. Es sencillamente la misma tarea de apertura de un fichero en BASIC por medio de OPENOUT, OPENIN o bien OPENUP. También sirve para cerrarlo, a semejanza de CLOSE# en BASIC. El registro A especifica la operación a realizar y los registros X e Y apuntan al nombre del fichero almacenado en memoria.

- *A=0:* Significa que se ha de cerrar el fichero. El registro Y retiene aquí el "número" atribuido al fichero cuando éste se abrió. Si se pone el número del fichero a 0 se cerrarán todos los ficheros actualmente abiertos.
- *A=&40:* Equivale a la orden OPENIN en BASIC. Se abre un fichero sólo para la grabación de datos (bytes).
- *A=&80:* Lo mismo que OPENOUT del BASIC. Se abre un fichero sólo para lectura de datos (bytes).
- *A=&C0:* Equivale al BASIC OPENUP. Esta vez se abre un fichero tanto para escribir como para leer en él. Se puede aplicar a cualquier sistema de ficheros que admita ficheros de acceso al azar.

Al usar OSFIND para abrir un fichero puede que no siempre tengamos éxito. Es posible que haya, por ejemplo, una etiqueta de protección contra escritura en el disco. Si el fichero no puede abrirse, el registro A se pone a cero al volver de OSFIND. Si tenemos éxito y el fichero es abierto, el "número" del fichero queda en A. Este valor se guardará para futuros usos.

Las siguientes llamadas a la ROM de un sistema de ficheros que examinaremos tienen algo en común: todas emplean el número del fichero generado por OSFIND.

La llamada OSARGS

Esta rutina es llamada en la dirección &FFDA y realiza numerosas tareas. Presenta la particularidad de que su bloque de parámetros, tan sólo de cuatro bytes de longitud, se almacena en la página cero. El registro X retiene la dirección del primer byte del bloque de parámetros, el Y retiene el número del fichero y el A especifica la operación que ha de ejecutarse. Existe un caso especial de OSARGS, que es cuando Y está a cero. En este caso la rutina da información sobre el sistema de ficheros empleado más que sobre el fichero en sí, o bien se encarga de que todos los ficheros abiertos en el sistema estén actualizados en relación a los datos que puedan quedar todavía en los buffers destinados a dichos ficheros. Veamos estas dos rutinas primero.

- *Y=0:A=0:* Una llamada OSARGS con estos parámetros y el registro X apuntando al área libre de la página cero produce una información del sistema de ficheros. Los datos se darán en el registro A. Esta llamada OSARGS es la única que funciona en el sistema para cassette. El dato que se da en el registro A se interpreta según el cuadro que ofrecemos al margen.
- *Y=0:A=255:* La otra llamada con Y a cero se produce cuando A retiene el valor 255. Una llamada a OSARGS en estas circunstancias hará que cada archivo quede actualizado con los buffers de la máquina. Si no hay ningún archivo abierto, no tendrá lugar actualización alguna.
- *Y=Número archivo:* Si el registro Y contiene el número del archivo, las operaciones realizadas dependen del contenido de A (como resumimos en el cuadro inferior). El "puntero secuencial" de un fichero abierto indica al sistema operativo dónde se ha de leer o escribir el siguiente byte. Este puntero resulta modificado por la función en BASIC PTR# y por estas rutinas. Como era de esperar, el dato se almacena en el bloque de parámetros de la página cero con su BmnS en la dirección (X+0). Cuando el dato es leído desde el sistema de ficheros por OSARGS, se almacena en el bloque de parámetros para su acceso posterior.

Contenido del reg. A	Descripción
0	Ningún sist. fich.
1	Cinta 1200 baud.
2	Cinta 300 baud.
3	ROM
4	DISCO
5	Red
6	Telesoft

Contenido del registro A	Descripción
0	Lee la posición actual del puntero secuencial en el fichero indicado por el registro Y
1	Escribe el valor del bloque de parámetros en el puntero secuencial. Se parece a la instr. PTR≠n=valor
2	Pone la longitud del archivo en el bloque de parámetros
255	Actualiza arch. en el soporte (el núm. de aquél está contenido en el reg. Y)

De esta exposición de la llamada resultará claro por qué el bloque de parámetros de OSARGS debe estar en la página cero: no queda más que un registro, el X, para guardar la dirección.

OSBGET y OSBPUT

Estas rutinas se usan, respectivamente, para leer y escribir bytes individuales en un fichero abierto. OSBGET se llama en la dirección &FFD7 y su empleo es muy sencillo. El registro Y se carga con el número de fichero, proporcionado por OSFIND, y llamando a &FFD7 se carga un byte. Éste es tomado (leído) de la posición actual del puntero secuencial y devuelto en A.

OSBPUT es la operación inversa; escribe un byte en el fichero en la posición indicada por el puntero secuencial. Para usar OSBPUT, el byte que se ha de escribir en el fichero se sitúa en A, mientras que Y retiene el número del fichero. Seguidamente se hace una llamada a &FFD4 para que ejecute la llamada. La breve rutina siguiente muestra el empleo de OSBPUT para grabar en un fichero un byte individual.

```
LDA   #&80     / preparación para abrir fichero
LDX   # name   MOD 256
LDY   # name   DIV 256
JSR   OSFIND   / abre fichero
STA   &71      / almacena número fichero
TAY            / lo pone además en el reg Y
LDA   #65      / byte a escribir
JSR   OSBPUT   / lo hace
LDA   #0
LDY   &71      / preparación para cerrar fichero
JSR   OSFIND   / cierra fichero
RTS            / fin
```

La llamada OSGBPB

Se llama en la dirección &FFD1. Transfiere grupos de bytes entre la memoria y los ficheros abiertos. Nos puede proporcionar también información sobre el sistema de ficheros en disco, si se está empleando. Aquí sólo nos ceñiremos, sin embargo, a su utilidad en la transferencia de datos. Y a este propósito diremos que la llamada no funciona en un sistema de ficheros en cinta; lo que no debe sorprender demasiado, dado que su utilidad es relevante sólo cuando empleamos ficheros de acceso directo desde lenguaje máquina.

El bloque de los parámetros para la llamada es el que se muestra en el siguiente cuadro, y se apunta a él por medio de los contenidos de X e Y, como es habitual. El registro A especifica la operación a realizar.

Bloque de parámetros para OSGBPB

Dir.	Descripción
0	Núm. fichero, proporcionado por OSFIND
1	BmnS de la dir. del dato por leer o escribir
2	Sin especificar
3	Sin especificar
4	BmsS de la dirección del dato
5	BmnS del número de bytes por trasladar
6	Sin especificar
7	Sin especificar
8	BmsS del número de bytes por trasladar
9	BmnS de la posición del puntero secuencial que hay que usar (permite la selección del lugar donde escribiremos o leeremos dentro del fichero)
10	Sin especificar
11	Sin especificar
12	BmsS de la posición del puntero secuencial

Una vez más advertimos que si usted emplea un BBC Micro estándar, la dirección se especifica por medio del contenido de los bytes 1 y 4 del bloque de parámetros.

Existen cuatro tipos de operaciones de transferencia de datos: dos de lectura y dos de escritura. Una operación de escritura y otra de lectura dan acceso a la información en el fichero en la posición especificada por la entrada del puntero secuencial en el bloque de parámetros mencionado anteriormente, mientras que las otras operaciones de lectura y escritura ignoran esta información por completo y utilizan el puntero secuencial en cualquier posición que se halle dentro del fichero. Así A=1 escribe bytes en el puntero especificado en el bloque; A=2 escribe bytes en el fichero ignorando la entrada del bloque de parámetros; A=3 lee datos del puntero secuencial especificado en el bloque; y A=4 lee bytes pero ignora el bloque de parámetros. El siguiente programa emplea las llamadas OSGBPB para escribir y leer un simple fichero de datos:

```
 10 REM solo para sistema en disco
 20 DIM block 20
 30 DIM data 100
 40 $data="Este es un programa de prueba"
 50 block!1=data  :  REM direccion del dato a escribir
 60 block!5=100:REM cantidad de datos a escribir
 70 block!9=0:REM puntero secuencial=0
 80 PRINT "Apertura fichero para lectura"
 90 Y%=OPENOUT("FICHERO") : REM abre el fichero
100 block?0=Y%               : REM numero de fichero
110 X%=block MOD 256
120 Y%=block DIV 256
130 A%=1: REM preparacion para escribir datos
140 CALL &FFD1 : REM lo hace
150 CLOSE £(block?0)
160 PRINT "Cierre del fichero"
170 TIME=0 : REPEAT:UNTIL TIME=400
180 PRINT "Apertura fichero para escritura"
190 PRINT "Ahora se leen los datos!"
200 $data=STRING$(100," "):REM borra area datos
210 Y%=OPENIN("FICHERO"): REM abre el fichero
220 block?0=Y%             : REM numero de fichero
230 block!1=data : REM direccion de datos para la operacion de lectura
240 block!5=100 : REM numero de bytes
250 block?9=0 : REM puntero secuencial a cero
260 X%=block MOD 256
270 Y%=block DIV 256
280 A%=3 : REM preparacion para lectura
290 CALL &FFD1 : REM lo hace
300 CLOSE£(block?0)
310 PRINT "Cierra de nuevo el fichero"
320 PRINT $data
330 END
```

El estado del flag de arrastre después de la llamada a &FFD1 indica si se ha logrado la transferencia o no. Si C está a cero entonces la transferencia se realizó con toda normalidad. Si C está a uno, es que ésta falló.

Otra llamada OS que, por último, tiene que ver con operaciones de ficheros es la OSFSC, pero su utilización es mínima a la hora de programar. La llamada es diferente de las otras ya mencionadas aquí dado que no existe una dirección de llamada directa para ella. Hay que llamarla en la dirección contenida en su vector. En los programas en lenguaje máquina, esta llamada puede hacerse con la instrucción

```
JMP     (&21E)
```

Hay otras rutinas OS que cumplen funciones específicas para ciertos sistemas de ficheros. Por ejemplo, las llamadas OSWORD se usan para acceder al chip de control de discos flexibles en un sistema de ficheros en disco. Igualmente, las tareas sencillas del sistema son confiadas a un pareja de llamadas OSBYTE.

Digitaya

En este capítulo de nuestra serie dedicada a programar juegos de aventuras damos el listado completo de "Digitaya"

La estructura de *Digitaya* guarda muchas semejanzas con la de *El bosque encantado*, pero el juego está hecho a una escala mucho mayor. El trazado original para *Digitaya* implicaba un mapa con 100 escenarios que conformaban el trazado interno de un ordenador, junto con memoria, bus de datos, procesador y gran parte del resto del hardware que hay en el interior de un ordenador personal medio. La tarea del jugador consiste en seguir el rastro del misterioso Digitaya, que está cautivo en algún lugar dentro de la máquina. Debe sortear muchos peligros, valiéndose de sus conocimientos sobre el trazado interno de un ordenador para rescatar a Digitaya.

Digitaya utiliza rutinas similares a las de *El bosque encantado* para llevar a cabo las funciones normales del juego, como trasladarse de un escenario a otro, recoger y dejar objetos y examinar los alrededores. No obstante, al esqueleto de esta estructura se le han añadido rutinas especiales para dar cabida a los peligros y trampas especiales del juego.

Cuando se participa en un juego de aventuras, suele ser aconsejable ir trazando un mapa de la ruta con lápiz y papel a medida que uno va atravesando el mundo de la aventura. La tarea primordial de todo jugador de aventuras que se precie consiste en ponerse en el lugar del creador del juego. Recrear el mapa de escenarios a menudo es un buen paso para alcanzar este objetivo. *Digitaya* está diseñado en una cuadrícula plana, pero no espere que todos los mapas de aventuras existan solamente en dos dimensiones. Con la clase de estructura de programa que hemos esbozado en el proyecto, son bastante factibles los mapas tridimensionales. También se podría concebir un juego en cuatro dimensiones, siendo el tiempo la cuarta dimensión. Tal juego, construido en base a un mapa en continuo cambio, probablemente derrotaría a todos los participantes excepto a los más experimentados.

El listado que ofrecemos es para el Commodore 64, aunque el programa se podrá ejecutar en la mayoría de las máquinas con BASIC tipo Microsoft. Se ofrecen complementos para el BBC Micro.

Se pueden suprimir del listado algunas sentencias REM para reducir el esfuerzo de introducción, pero para asegurar que el programa funcione correctamente suprima sólo las REM del final de las líneas de programa que incluyan otro código. Las líneas que sólo contienen sentencias REM por lo general son títulos de subrutinas y sus números de línea se utilizan para las llamadas GOSUB. Es probable que la eliminación de tales líneas provoque un error UNDEFINED STATEMENT (sentencia no definida).

Complementos al BASIC

Spectrum:

Debido a la forma inusual en que el Spectrum manipula las matrices y variables en serie, los complementos para *Digitaya* son tan extensos que sería imposible listarlos aquí. Usted habrá de remitirse a los capítulos anteriores. No obstante, para ayudarlo a recodificar el listado, detallamos los puntos principales a los que atender. Primero, el Spectrum sólo permite variables en serie de un único carácter. Todos los nombres de variables que requieran conversión se ofrecieron ya en una tabla. Asimismo, el Spectrum sólo admite matrices en serie de longitud fija, estableciéndose la longitud de cada elemento de la matriz mediante la sentencia DIM. Ello puede plantear problemas si, pongamos por caso, se dimensionara una matriz en serie de modo que cada elemento tuviera 20 caracteres de largo. Si a un elemento de la matriz se le asignara entonces una serie de 15 caracteres, los cinco caracteres restantes del elemento se rellenarían con espacios en blanco. Hemos de recortar los espacios del elemento de la matriz, por ejemplo, antes de añadírsela a cualquier frase. La variable A$ se emplea para pasar el elemento a una subrutina y debe ser asignada antes de efectuar la llamada a la subrutina. He aquí un ejemplo:

Versión Microsoft:

```
3650 SN$="TU  "+IV$(F,1)+"NO SIRVE DE
      NADA, LA FUERZA SE INTENSIFICA"
```

Versión para el Spectrum:

```
3650 LET S$="TU  ":A$=IV$(F,1):GOSUB
      8500:LETS$=S$+" NO SIRVE DE
      NADA, LA FUERZA SE INTENSIFICA"
```

El otro problema es el borrado de la pantalla. En la versión Commodore se consigue mediante PRINT CHR$(147). Simplemente reemplace estas sentencias por CLS.

BBC Micro:

En el listado de *Digitaya* reemplace LNS() por LS() y LN por L. Sustituya, además, las siguientes líneas:

```
1400 A$=GET$
1410 CLS
2630 REPEAT:A$=GET$:UNTIL A$="S" OR
     A$="N"
2750 RA=RND(1)
2820 P=RND(40)+7
3890 RD=RND(1):IF RD>.65 THEN 4110:REM
     ACERTADO
4090 P=RND(40)+7
4520 IV$(4,2)=STR$(RND(40)+7):REM
     REASIGNAR POSICION BILLETE
4570 RN=RND(3)+1
5560 RA=RND(1)
```

La aventura de Digitaya

```
1030 REM **          "DIGITAYA"          **
1040 REM ** UN JUEGO DE AVENTURAS **
1050 REM **      PARA ORDENADOR          **
1090 :
1100 GOSUB6090:REM LEER DATOS MATRIZ
1110 GOSUB1290:REM HISTORIA HASTA AHORA
1120 P=47:REM PUNTO DE PARTIDA
1130 :
1140 REM REM **** AQUI COMIENZA EL BUCLE PRINCIPAL ****
1150 :
1160 MF=0:PRINT
1170 GOSUB1440:REM DESCRIBIR POSICION
1180 GOSUB1560:REM LISTAR SALIDAS
1190 GOSUB 2670:REM ES ESPECIAL P
1200 IF SF=1 THEN 1250:REM SIGUIENTE BUCLE
1210 PRINT:INPUT"INSTRUCCIONES";IS$
1220 GOSUB1700:REM ANALIZAR INSTRUCCIONES
1225 IF F=0 THEN 1210:REM INSTRUCCION NO VALIDA
1230 GOSUB 1900:REM INSTRUCCIONES NORMALES
1240 IF VF=0 THENPRINT"NO COMPRENDO"
1250 IF MF=1 THEN 1160:REM NUEVA POSICION
1260 IF MF=0 THEN 1210:REM NUEVA INSTRUCCION
1270 END
1280 :
1290 REM **** HISTORIA HASTA AHORA ****
1300 SN$="BIENVENIDO A 'DIGITAYA'"
1310 GOSUB5880:REM FORMATEAR
1320 PRINT
1330 SN$="MIENTRAS LA MAQUINA ZUMBA QUEDAMENTE, MIRAS A TU ALREDEDOR."
1340 SN$=SN$+"  HACIA EL NORTE Y HACIA EL SUR SE EXTIENDE UN ANCHO BUS."
1350 SN$=SN$+"  TU MISION CONSISTE EN ENCONTRAR AL MISTERIOSO DIGITAYA"
1360 SN$=SN$+"  Y PONERLO A SALVO SACANDOLO A TRAVES DE UNA DE LAS PUERTAS DE
     SALIDA."
1370 SN$=SN$+" ... PERO, DE CUAL?"
1380 GOSUB 5880
1390 PRINT:PRINT"PULSA UNA TECLA PARA EMPEZAR"
1400 GETA$:IFA$=""THEN1400
1410 PRINTCHR$(147):REM LIMPIAR PANTALLA
1420 RETURN
1430 :
1440 REM **** S/R DESCRIBIR POSICION ****
1450 SN$="TE HALLAS EN  "+LN$(P):GOSUB5880
1460 SN$="VES  "
1470 REM ** BUSCAR OBJETO **
1480 F=0:SP$=""
1490 FOR I=1 TO 8
1500 IF VAL(IV$(I,2))=P  THEN SN$=SN$+SP$+"UN  "+IV$(I,1):F=1:SP$="    "
1510 NEXT I
1520 IF F=0 THEN SN$=SN$+"NINGUN OBJETO"
1530 GOSUB5880:REM FORMATEAR
1540 RETURN
1550 :
1560 REM **** S/R LISTAR SALIDAS ****
1570 EX$=EX$(P)
1580 NR=VAL(LEFT$(EX$,2))
1590 EA=VAL(MID$(EX$,3,2))
1600 SO=VAL(MID$(EX$,5,2))
1610 WE=VAL(RIGHT$(EX$,2))
1620 IF(NR OR EA OR SO OR WE)=0THEN RETURN
1630 PRINT:SN$="HAY SALIDAS POR EL  "
1640 IF NR<>0 THEN SN$=SN$+"NORTE  "
1650 IF EA<>0 THEN SN$=SN$+ "ESTE"
1660 IF SO<>0 THEN SN$=SN$+ "SUR"
1670 IF WE<>0 THEN SN$=SN$+ "OESTE"
1675 GOSUB5880:REM FORMATEAR
1680 PRINT:RETURN
1690 :
1700 REM **** S/R ANALIZAR INSTRUCCION ****
1705 F=0:REM PONER BANDERA A CERO
1710 IFIS$="FIN" OR IS$="LISTAR" THEN VB$=IS$:F=1:RETURN
1720 IF IS$="MIRAR"   THEN VB$=IS$:F=1:RETURN
1730 :
1740 REM ** DESCOMPONER INSTRUCCION **
1750 VB$="":NN$="":REM PONER A CERO VERBO Y SUSTANTIVO
1770 LS=LEN(IS$)
1780 FOR   C=1 TO LS
1790 A$=MID$(IS$,C,1)
1800 IF A$="  " THEN VB$=LEFT$(IS$,C-1):NN$=RIGHT$(IS$,LS-C):F=
     1:C=LS
1810 NEXT
1830 IF F=0 THEN PRINT:PRINT"NECESITO AL MENOS DOS PALABRAS"
1840 RETURN
1850 :
1900 REM **** S/R ACCIONES NORMALES ****
1910 VF=0
1920 PRINT
1930 IF VB$="IR" OR VB$="AVANZAR"THEN VF=1:GOSUB2000
1940 IF VB$="RECOGER" ORVB$="COGER" THEN VF=1:GOSUB2140
1950 IF VB$="DEJAR" ORVB$="PONER" THENVF=1:GOSUB2360
1960 IF VB$="LISTAR"ORVB$="INVENTARIO"THEN VF=1:GOSUB2540
1965 IF VB$="MIRAR" THEN VF=1:MF=1:RETURN
1970 IF VB$="FIN"ORVB$="TERMINAR" THEN VF=1:GOSUB2610
1980 RETURN
1990 :
2000 REM **** S/R DE MOVIMIENTO ****
2010 MF=1:REM ESTABLECER BANDERA DE MOVIMIENTO
2015 GOSUB8600:REM BUSCAR DIRECCION
2020 DR$=LEFT$(NN$,1)
2030 IFDR$<>"N"ANDDR$[illegible]"S"ANDDR$<>"0"THEN2100
2040 IF DR$="N" AND NR<>0 THEN P=NR:RETURN
2050 IF DR$="S" AND SO<>0 THEN P=SO:RETURN
2060 IF DR$="E" AND EA<>0 THEN P=EA:RETURN
2070 IF DR$="0" AND WE<>0 THEN P=WE:RETURN
2080 PRINT"NO PUEDES  ";IS$
2090 MF=0:RETURN
2100 REM EL NOMBRE NO ES OK
2110 PRINT"QUE ES  ";NN$;"  ?"
2120 MF=0:RETURN
2130 :
2140 REM **** S/R RECOGER ****
2145 IV$(4,1)="BILLETE AL TRIESTADO"
2150 GOSUB5730:REM ES VALIDO EL OBJETO
2160 IF F=0 THEN PRINT"NO HAY NINGUN  ";W$:RETURN
2170 REM ** YA HA RECOGIDO EL OBJETO ? ****
2180 OV=F:GOSUB5830
2190 IFHF=1 THEN SN$="YA TIENES EL  "+IV$(F,1):GOSUB5880:RETURN
2200 :
2210 REM ** ESTA AQUI EL OBJETO **
2220 IF VAL(IV$(F,2))<>P THEN SN$=IV$(F,1)+"NO ESTA AQUI":GOSUB5880: RETURN
2230 :
2240 REM ** AÑADIR OBJETO A LA LISTA **
2250 AF=0:FOR J=1 TO 4
2260 IFIC$(J)=""THENIC$(J)=IV$(F,1):AF=1:J=4
2270 NEXTJ
2280 :
2290 REM ** COMPROBAR SI CUOTA CUBIERTA **
2300 IF AF=0THENPRINT"YA TIENES 4 OBJETOS":RETURN
2310 :
2320 SN$="COGES EL  "+IV$(F,1):GOSUB5880
2330 IV$(F,2)="-":REM SUPRIMIR ENTRADA POSICION
2340 RETURN
2350 :
2360 REM ** S/R DEJAR **
2370 GOSUB5730:REM ES VALIDO EL OBJETO
2380 IF F=0 THEN PRINT"NO HAY NINGUN  ";NN$:RETURN
2390 :
2400 REM ** LLEVA CONSIGO EL OBJETO ? **
2410 OV=F:GOSUB5830
2420 IFHF=0THENPRINT"TU NO TIENES EL  ";IV$(F,1):RETURN
2430 :
2440 REM ** DEJAR OBJETO **
2450 SN$="DEJAS EL  "+IV$(F,1):GOSUB5880
2460 IV$(F,2)=STR$(P):REM ACTUALIZAR POSICION OBJETO
2470 :
2480 REM ** SUPRIMIRLO DE LISTA DE OBJ TRANSPORTADOS **
2490 FOR J=1T04
2500 IF IC$(J)=IV$(F,1)THEN IC$(J)="":J=4
2510 NEXTJ
2520 RETURN
2530 :
2540 REM **** S/R LISTAR INVENTARIO ****
2550 PRINT"OBJETOS QUE LLEVAS CONTIGO:"
2560 FORI=1T04
2570 PRINT"    ";IC$(I)
2580 NEXTI
2590 RETURN
2600 :
2610 REM **** S/R FIN DEL JUEGO ****
2620 PRINT:PRINT"ESTAS SEGURO (S/N) ?"
2630 GETA$:IFA$<> "S" AND A$<>"N"THEN2630
2640 IFA$="N"THEN RETURN
2650 END
2660 :
2670 REM **** S/R ES ESPECIAL P ****
2680 SF=0:REM QUITAR BANDERA DE ESPECIAL
2690 IF P=37 THEN2780:REM TABLA DE VECTORES
2700 IF P>7 THEN 2750:REM BICHO AL AZAR
2710 ON P GOSUB 2850,2960,3450,3830,4180,4550,5150
2720 RETURN
2730 :
2740 REM ** BICHO AL AZAR **
2750 RA=RND(TI)
2760 IF RA<0.05 THEN GOSUB 5420:REM BICHO
2770 RETURN
2780 REM ** TABLA DE VECTORES **
2790 SF=1
2800 SN$="AVANZAS A GRAN VELOCIDAD HASTA UN NUEVO ESCENARIO":GOSUB5880
2810 FORJ=1TO1000:NEXT:REM PAUSA
2820 P=INT(RND(TI)*40+7)
2830 MF=1:RETURN
2840 :
2850 REM **** S/R TOMA DEL TELEVISOR ****
2860 SF=1
2870 SN$="HAS ENTRADO EN LA TOMA DEL TELEVISOR Y NO TIENES SALIDA."
2880 SN$=SN$+"ESTAS CONDENADO PARA SIEMPRE A SER EL INVITADO DE UN PROGRAMA DE
     TERTULIA DE TV"
2890 GOSUB5880:REM FORMATEAR SALIDA
2900 PRINT
2910 PRINT"BIENVENIDOS AL SHOW......."
2920 FORJ=1TO500:NEXTJ
2930 GOTO 2910
2940 END
2950 :
2960 REM **** S/S PUERTA PARA EL USUARIO ****
2970 SF=1
2980 SN$="PUEDES ESCAPAR PERO EL VENDEDOR DE BILLETES DEL RDD"
2990 SN$=SN$+"  TE IMPIDE EL PASO. TE DICE QUE TIENE INSTRUCCIONES"
3000 SN$=SN$+"  DE ACEPTAR SOLO ENTRADAS. SIN EMBARGO ACEPTA TODAS LAS
     PRINCIPALES"
```

```
3010 SN$=SN$+ "TARJETAS DE CREDITO."
3020 GOSUB 5880:REM FORMATEAR IMPRESION
3030 :
3040 PRINT:INPUT"INSTRUCCIONES";IS$
3050 GOSUB1700:REM ANALIZAR INSTRUCCIONES
3060 GOSUB1900: REM ACCIONES NORMALES
3070 IF MF=1 THEN RETURN:REM IRSE
3080 IF VF=1 THEN3040:REM SIGUIENTE INSTRUCCION
3090 IF VB$<>"DAR" THENPRINT"NO COMPRENDO":GOTO3040
3100 :
3110 REM ** LA INSTRUCCION ES DAR **
3120 GOSUB5730:REM ES VALIDO EL OBJETO
3130 IFF=0THENPRINT"NO HAY NINGUN ";NN$:GOTO3040:REM SIGUIENTE INSTRUCCION
3140 :
3150 REM ** ES EL OBJ TARJETA DE CREDITO **
3160 IF F<>5 THEN PRINT"EL SOLO ACEPTA TARJETAS DE CREDITO":GOTO3040
3170 :
3180 REM ** LLEVA LA TARJETA **
3190 OV=5:GOSUB5830
3200 IFHF=0THENPRINT"TU NO TIENES LA ";IV$(5,1):GOTO 3040
3210 :
3220 SN$="EL VENDEDOR COGE LA TARJETA Y DICE 'ESTA SERVIRO, SEÑOR'"
3230 GOSUB5880:REM FORMATEAR SALIDA
3240 SN$="SE TE PERMITE ATRAVESAR LA BARRERA Y ENTRAR EN LA PUERTA PARA EL
     USUARIO"
3250 GOSUB5880:REM FORMATEAR SALIDA
3260 :
3270 REM ** LLEVA A DIGITAYA **
3280 OV=6:GOSUB5830
3290 IF HF=1 THEN 3380:REM EXITO
3300 :
3310 REM ** FRACASO **
3320 SN$="BIEN HECHO.HAS CONSEGUIDO ESCAPAR DE LAS GARRAS"
3330 SN$=SN$+" DE LA MAQUINA, PERO HAS FRACASADO EN TU MISION"
3340 SN$=SN$+" DE TRAER DE REGRESO AL MISTERIOSO DIGITAYA"
3350 GOSUB5880:REM FORMATEAR SALIDA
3360 END
3370 :
3380 REM ** EXITO **
3390 SN$="FELICITACIONES, HAS TENIDO EXITO EN TU MISION"
3400 SN$=SN$+" CONSISTENTE EN RESCATAR AL MARAVILLOSO DIGITAYA DE LAS"
3410 SN$=SN$+" GARRAS DE LA MAQUINA."
3420 GOSUB5880:REM FORMATEAR SALIDA
3430 END
3440 :
3450 REM **** S/R PUERTA PARA CASSETTE ****
3460 SF=1
3470 SN$="SIENTES QUE UNA FUERZA IRRESISTIBLE TE EMPUJA HACIA"
3480 SN$=SN$+" LA SUSPENSION MAGNETICA PERMANENTE"
3490 GOSUB5880:REM FORMATEAR
3500 NS=0:REM COMENZAR A CONTAR INSTRUCCIONES
3510 REM ** INSTRUCCIONES **
3520 NS=NS+1:IFNS>3THEN3770:REM ABSORBIDO
3530 PRINT:INPUT"INSTRUCCIONES";IS$
3540 GOSUB1700:REM ANALIZAR INSTRUCCIONES
3550 GOSUB1900:REM ACCIONES NORMALES
3560 IF MF=1 THEN MF=0:PRINT"NO PUEDES MOVERTE...TODAVIA":GOTO3510
3570 IFVF=1THEN3510:REM SIGUIENTE INSTRUCCION
3580 IFVB$<>"USAR"THENPRINT"NO COMPRENDO":GOTO3510
3590 REM ** LA INSTRUCCION ES USAR **
3600 GOSUB5730:REM ES VALIDO EL OBJETO
3610 IFF=0THENPRINT"NO HAY NINGUN ";NN$:GOTO3510
3620 :
3630 REM ** ES EL OBJETO EL ACTIVADOR DEL BUFFER **
3640 IF F=8 THEN 3680:REM OK
3650 SN$="TU "+IV$(F,1)+" NO SIRVE DE NADA, LA FUERZA AUMENTA"
3660 GOSUB 5880:GOTO3510:REM SIGUIENTE INSTRUCCION
3670 :
3680 OV=8:GOSUB5830:REM LLEVA EL ACT DEL BUFFER
3690 IF HF=0THENSN$="NO LLEVAS CONTIGO EL "+IV$(8,1):GOSUB5880:GOTO 3510
3700 :
3710 REM ** SALVADO **
3720 SN$="USAS EL ACTIVADOR DEL BUFFER PARA CONTRARRESTAR EL EMPUJE"
3730 SN$=SN$+" HACIA EL OLVIDO MAGNETICO. LA FUERZA DISMINUYE"
3740 GOSUB5880:REM FORMATEAR
3750 RETURN
3760 :
3770 REM ** ABSORBIDO **
3780 SN$="LA FUERZA SE HACE DEMASIADO INTENSA Y ERES ARROJADO"
3790 SN$=SN$+" A TRAVES DE LA PUERTA PARA CASSETTE A LA NADA MAGNETICA."
3800 GOSUB5880:REM FORMATEAR
3810 END
3820 :
3830 REM **** PUERTA PARA PALANCA DE MANDOS ****
3840 SF=1
3850 SN$="UN USUARIO DE OJOS ENROJECIDOS DISPARA REPETIDAMENTE SU LASER CONTRA
     TI."
3860 GOSUB5880:REM FORMATEAR
3870 :
3880 REM ** INSTRUCCIONES **
3890 RD=RND(TI):IF RD>.65 THEN 4110:REM ACERTADO
3900 PRINT:INPUT"INSTRUCCIONES";IS$
3910 GOSUB1700:GOSUB1900:REM ANALIZAR INSTRUCCION
3920 IFMF=1THENMF=0:PRINT"NO PUEDES MOVERTE...TODAVIA":GOTO3880
3930 IFVF=1THEN3880:REM SIGUIENTE INSTRUCCION
3940 IF VB$<>"USAR" THENPRINT"NO COMPRENDO":GOTO3880
3950 GOSUB5730:REM ES VALIDO EL OBJETO
3960 IFF=0THENPRINT"NO HAY NINGUN ";NN$:GOTO3880:REM SIGUIENTE INSTRUCCION
3970 :
3980 REM ** ES EL OBJETO EL ESCUDO LASER **
3990 IF F=3 THEN4020:REM OK
4000 SN$="TU "+IV$(F,1)+" NO SIRVE":GOSUB5880:GOTO3880
4010 :
4020 OV=3:GOSUB5830:REM LLEVA CONSIGO EL ESCUDO LASER
4030 IF MF=0 THENSN$="TU NO TIENES EL "+IV$(3,1):GOSUB5880:GOTO3880
4040 :
4050 REM ** SALVADO **
4060 SN$="USAS EL ESCUDO LASER PARA PROTEGERTE, UNA DESCARGA TE ARROJA"
4070 SN$=SN$+" FUERA DE LA PUERTA PARA PALANCA DE MANDOS Y VUELVES A ENTRAR EN
     LA MAQUINA."
4080 GOSUB5880:REM FORMATEAR
4090 P=INT(RND(TI)*40+7):MF=1:RETURN
4100 :
4100 REM ** ACERTADO **
4120 SN$="ERES ALCANZADO POR EL LASER Y APENAS SI ERES CONSCIENTE DE"
4130 SN$=SN$+" QUE TUS ATOMOS SE HAN DISPERSADO POR LAS CUATRO ESQUINAS"
4140 SN$=SN$+" DEL UNIVERSO"
4150 GOSUB5880:REM FORMATEAR
4160 END
4170 :
4180 REM **** S/R DISPOSITIVO TRIESTADO ****
4190 SF=1
4200 SN$="UN GRAN CARTEL REZA 'E/S POR AQUI' PERO CUANDO TE ACERCAS"
4210 SN$=SN$+" UN REVISOR GRITA 'BILLETES POR FAVOR'"
4220 GOSUB5880:REM FORMATEAR
4230 :
4240 REM ** INSTRUCCIONES **
4250 PRINT:INPUT"INSTRUCCIONES";IS$
4260 GOSUB1700:GOSUB1900:REM ANALIZAR
4270 IFMF=1 THEN RETURN
4280 IFVF=1THEN4240:REM SIGUIENTE INSTRUCCION
4290 IFVB$<> "DAR" AND VB$<>"OFRECER"THENPRINT"NO COMPRENDO":GOTO4240
4300 REM ** LA INSTRUCCION ES DAR **
4310 GOSUB5730:REM ES VALIDO EL OBJETO
4320 IFF=0THENPRINT"NO HAY NINGUN ";NN$:GOTO4240:REM SIGUIENTE INSTRUCCION
4330 :
4340 REM ** ES EL OBJETO EL BILLETE **
4350 IF F=4 THEN 4400:REM OK
4360 SN$="EL REVISOR SACUDE LA CABEZA Y DICE"
4370 SN$=SN$+" 'NO PUEDO ACEPTAR ESTO "+IV$(F,1)
4380 GOSUB5880:GOTO4240:REM SIGUIENTE INSTRUCCION
4390 :
4400 OV=4:GOSUB5830:REM LLEVA CONSIGO EL BILLETE
4410 IFHF=0THENPRINT"TU NO TIENES EL BILLETE":GOTO 4240
4420 :
4430 REM ** OK **
4440 SN$="EL REVISOR ACEPTA TU BILLETE Y TE PERMITE"
4450 SN$=SN$+" ATRAVESAR LA BARRERA."
4460 GOSUB5880:REM FORMATEAR
4470 REM ** SUPRIMIR BILLETE DE LA LISTA **
4480 F=0
4490 FORJ=1TO4
4500 IF IC$(J)=IV$(4,1)THENIC$(J)="":J=4
4510 NEXT J
4520 IV$(4,2)=STR$(INT(RND(TI)*40+8)):REM REASIGNAR POSICION BILLETE
4530 P=15:MF=1:RETURN
4540 :
4550 REM **** ALU ****
4560 SF=1
4570 RN=INT(RND(TI)*3+1)
4580 IF RN=1 THEN CD$="AND"
4590 IF RN=2 THEN CD$="OR"
4600 IF RN=3 THEN CD$="NOT"
4610 SN$="MONTADOS EN LA PARED HAY TRES BOTONES MARCADOS"
4620 SN$=SN$+" 'AND', 'OR' Y 'NOT'. PUEDES GANAR ACCESO AL"
4630 SN$=SN$+" ACUMULADOR PULSANDO EL BOTON CORRECTO"
4640 GOSUB5880:REM FORMATEAR
4650 :
4660 REM ** INSTRUCCIONES **
4670 PRINT:INPUT"INSTRUCCIONES";IS$
4680 GOSUB1700:GOSUB1900:REM ANALIZAR
4690 IF MF=1 THEN RETURN:REM IRSE
4700 IF VF=1 THEN 4670:REM SIGUIENTE INSTRUCCION
4710 IF VB$="USAR" OR VB$="PULSAR" THEN4740
4720 PRINT"NO COMPRENDO":GOTO4670
4730 :
4740 REM ** INSTRUCCION VALIDA **
4750 IF VB$="PULSAR"THEN 4930
4760 REM ** LA INSTRUCCION ES USAR **
4770 GOSUB5730:REM ES VALIDO EL OBJETO
4780 IFF=0THENPRINT"NO HAY NINGUN ";NN$:GOTO4670:REM SIGUIENTE INSTRUCCION
4790 :
4800 REM ** ES EL OBJETO EL LIBRO DE CODIGOS **
4810 IF F=7 THEN 4850:REM OK
4820 SN$="TU "+ IV$(F,1)+" NO SIRVE":GOSUB5880
4830 GOTO4670:REM SIGUIENTE INSTRUCCION
4840 :
4850 OV=7:GOSUB5830:REM LLEVA CONSIGO EL LIBRO DE CODIGOS
4860 IFHF=1THEN4900:REM OK LO LLEVA
4870 SN$="TU NO TIENES EL "+IV$(7,1)
4880 GOSUB5880:GOTO4670:REM SIGUIENTE INSTRUCCION
4890 :
4900 SN$="ABRES EL LIBRO DE CODIGOS Y EN SU INTERIOR ENCUENTRAS ESCRITA LA
     PALABRA '"+CD$+"'"
4910 GOSUB5880:GOTO4670:REM SIGUIENTE INSTRUCCION
4920 :
4930 REM ** LA INSTRUCCION ES PULSAR **
4940 IFNN$="AND"OR NN$="OR"OR NN$="NOT" THEN 4970
4950 SN$="NO HAY NINGUN "+NN$:GOSUB5880:GOTO4670:REM SIGUIENTE INSTRUCCION
4960 :
4970 REM ** CORRECTO O INCORRECTO **
4980 IFNN$=CD$ THEN GOSUB5100:RETURN
```

Ian McKinnell

```
4990 GOSUB5010:RETURN
5000 :
5010 REM **** S/R INCORRECTO ****
5020 SN$="INCORRECTO, SE ABRE UNA PUERTA FALSA Y TE ENCUENTRAS OTRA"
5030 SN$=SN$+" VEZ EN LA MEMORIA PRINCIPAL"
5040 GOSUB5880:REM FORMATEAR
5050 IF RN=1 THEN P=39
5060 IF RN=2 THEN P=35
5070 IF RN=3 THEN P=29
5080 MF=1:RETURN
5090 :
5100 REM ** S/R CORRECTO **
5100 SN$="SE ABRE LA PUERTA QUE DA AL ACUMULADOR Y"
5120 SN$=SN$+" LA CRUZAS":GOSUB5880
5130 P=30:MF=1:RETURN
5140 :
5150 REM **** S/R PUERTA A LA MEMORIA ****
5160 SF=1
5170 SN$="TE SALUDA UN CONSERJE PERO TE DICE QUE NO TE PUEDE DEJAR PASAR"
5180 SN$=SN$+" A MENOS QUE LE DES UNA DIRECCION":GOSUB5880
5190 REM ** INSTRUCCIONES **
5200 PRINT:INPUT"INSTRUCCIONES";IS$
5210 GOSUB1700:GOSUB1900:REM ANALIZAR
5220 IF MF=1 THEN RETURN:REM IRSE
5230 IF VF=1 THEN 5200:REM SIGUIENTE INSTRUCCION
5240 IF VB$<>"DAR"THEN PRINT"NO COMPRENDO":GOTO5200
5250 :
5260 GOSUB5730:REM ES VALIDO EL OBJETO
5270 IFF=0THENPRINT"NO HAY NINGUN ";NN$:GOTO5200:REM SIGUIENTE INSTRUCCION
5280 :
5290 REM ** ES EL OBJETO DIRECCION **
5300 IF F=1 THEN5330:REM OK
5310 PRINT"NECESITA TU DIRECCION":GOTO5200
5320 :
5330 OV=1:GOSUB5830:REM LLEVA LA DIRECCION
5340 IF HF=1 THEN 5370
5350 SN$="TU NO TIENES LA "+IV$(1,1):GOSUB5880:GOTO5200
5360 :
5370 REM ** OK PASAR **
5380 SN$="EL CONSERJE EXAMINA TU DIRECCION Y TE PERMITE"
5390 SN$="SN$+" PASAR":GOSUB5880
5400 P=40:MF=1:RETURN
5410 :
5420 REM **** BICHO AL AZAR ****
5430 SF=1
5440 SN$="POR DETRAS DE UN CHIP APARECE UN BICHO ENORME Y ASQUEROSO"
5450 SN$="SN$+" Y ARREMETE CONTRA TI":GOSUB5880
5460 :
5470 REM ** INSTRUCCIONES **
5480 PRINT:INPUT"INSTRUCCIONES";IS$
5490 GOSUB1700:GOSUB1900:REM ANALIZAR
5500 IFMF=1THENMF=0:PRINT"NO PUEDES MOVERTE...TODAVIA":GOTO5480
5510 IFVF=1THEN5480:REM SIGUIENTE INSTRUCCION
5520 IF VB$="MATAR" OR VB$="LUCHAR" THEN5550
5530 PRINT"NO COMPRENDO":GOTO5480
5540 :
5550 REM ** LA INSTRUCCION ES LUCHAR/MATAR **
5560 RA=RND(TI)
5570 IFRA<0.5THENGOSUB5600
5580 GOSUB5670:RETURN
5590 :
5600 REM **** S/R MUERTO ****
5610 SN$="LUCHAS CON EL BICHO. TE ARROJA UNA LLUVIA DE ERRORES"
5620 SN$=SN$+" QUE SE TE METEN EN EL CEREBRO."
5630 SN$=SN$+" FINALMENTE TU CABEZA YA NO RESISTE MAS Y ESTALLA."
5640 GOSUB5880
5650 END
5660 :
5670 REM **** S/R LO MATAS ****
5680 SN$="LUCHAS CON EL BICHO Y AUNQUE LA PELEA ES DURA"
5690 SN$=SN$+" FINALMENTE LOGRAS VENCERLO Y SOBREVIVIR.":GOSUB5880
5700 RETURN
5710 :
5720 :
5730 REM **** S/R OBJETO VALIDO ****
5740 NN$=NN$+" ":LN=LEN(NN$):F=0:C=1
5745 FOR K=1 TO LN
5750 IF MID$(NN$,K,1)<>" " THENNEXT K:RETURN
5755 W$=MID$(NN$,C,K-C):C=K+1:LW=LEN(W$)
5760 FORJ=1 TO 8
5770 LI=LEN(IV$(J,1)):REM LONGITUD OBJETO
5780 FORI=1 TO LI
5790 IFMID$(IV$(J,1),I,LW)=W$THENF=J:I=LI:J=8:K=LN
5800 NEXT I,J,K
5810 RETURN
5820 :
5830 REM **** S/R LLEVA CONSIGO OBJETO ****
5840 MF=0
5850 IFIV$(OV,2)="-1"THEN HF=1
5860 RETURN
5870 :
5880 REM **** S/R FORMATEAR IMPRESION ****
5890 LC=0:     REM CONTADOR CAR/LINEA
5900 OC=1:     REM CONTADOR ANTIGUO
5910 OW$="":   REM PALABRA ANTIGUA
5920 LL=40:    REM LONGITUD LINEA PANTALLA
5930 SN$=SN$+" FICTICIA"
5940 PRINT
5950 FOR C=1 TO LEN(SN$)
5960 LC=LC+1
5970 IF MID$(SN$,C,1)=" " THENGOSUB6020
5980 NEXTC
5990 PRINT
6000 RETURN
6010 :
6020 REM **** S/R COMPROBACION FINAL LINEA ****
6030 NW$=MID$(SN$,OC,C-OC+1)
6040 IF LC<LL THENPRINTOW$;:GOTO6060
6050 PRINTOW$:LC=LEN(NW$)
6060 OC=C+:OW$=NW$
6070 RETURN
6080 :
6090 REM **** S/R LEER DATOS MATRIZ ****
6100 REM ** LEER INVENTARIO **
6110 DIM IV$(8,2),IC$(4)
6120 FOR C=1TO8
6130 READ IV$(C,1),IV$(C,2)
6140 NEXT C
6150 :
6160 REM ** LEER DATOS ESCENARIO Y SALIDAS **
6170 DIM LN$(55),EX$(55)
6180 C1=0:C2=0:REM INICIALIZAR SUMAS DE CONTROL
6190 FOR C=1 TO 54
6200 READ LN$(C),EX$(C)
6210 C1=C1+VAL(LEFT$(EX$(C),4))
6220 C2=C2+VAL(RIGHT$(EX$(C),4))
6230 NEXT C
6240 READ CA:IF CA<>C1 THEN PRINT"ERROR EN SUMA DE CONTROL":STOP
6250 READ CB:IF CB<>C2 THEN PRINT"ERROR EN SUMA DE CONTROL":STOP
6260 RETURN
6270 REM **** DATOS INVENTARIO ****
6280 DATA NUMERO DE DIRECCION,45,LLAVE,34,ESCUDO LASER,25
6290 DATA BILLETE AL TRIESTADO,26,TARJETA DE CREDITO DATOS,28
6300 DATA DIGITAYA,30,LIBRO DE CODIGOS,19,DISPOSITIVO ACTIVADOR DEL BUFFER,13
6310 :
6320 REM **** DATOS ESCENARIOS Y SALIDAS ****
6330 DATA EN LA TOMA DEL TELEVISOR,00000000
6340 DATA EN LA PUERTA PARA EL USUARIO,00090100
6350 DATA EN LA PUERTA PARA CASSETTE,00110000
6360 DATA EN LA PUERTA PARA PALANCA DE MANDOS,00130000
6370 DATA EN UN DISPOSITIVO TRIESTADO,00170000
6380 DATA EN LA UNIDAD ARITMETICO LOGICA,00310016
6390 DATA EN LA PUERTA DE ACCESO A LA MEMORIA,00490000
6400 DATA EN EL BUS DE E/S,09000001
6410 DATA EN EL BUS DE E/S,10000802
6420 DATA EN EL BUS DE E/S,11000900
6430 DATA EN EL BUS DE E/S,12001003
6440 DATA EN EL BUS DE E/S,13531100
6450 DATA EN EL BUS DE E/S,14001204
6460 DATA EN EL BUS DE E/S,15001300
6470 DATA EN EL BUS DE E/S UN CARTEL REZA 'S OUT H',00001400
6480 DATA EN EL REGISTRO DE DATOS,00061700
6490 DATA EN UN BUS DE 8 PISTAS,16001805
6500 DATA EN UN BUS DE 8 PISTAS,17001900
6510 DATA EN UN BUS DE 8 PISTAS,18002000
6520 DATA EN EL BUS DE 8 PISTAS,19292100
6530 DATA EN EL BUS DE 8 PISTAS,20282200
6540 DATA EN EL BUS DE 8 PISTAS,21272300
6550 DATA EN UN BUS DE 8 PISTAS,22262400
6560 DATA EN UN BUS DE 8 PISTAS,23250000
6570 DATA EN LA MATRIZ DE CARACTERES,26360024
6580 DATA EN LA PARTE SUPERIOR DE LA MEMORIA,27352523
6590 DATA EN LA MITAD DE LA MEMORIA,28342622
6600 DATA EN LA MITAD DE LA MEMORIA,29332721
6610 DATA EN LA PARTE INFERIOR DE LA MEMORIA,00542820
6620 DATA EN LA GUARIDA DEL ACUMULADOR, 00000600
6630 DATA EN UN LARGO CORREDOR,00420006
6640 DATA EN UN REGISTRO DE INDICE,31000000
6650 DATA EN LA PARTE INFERIOR DE LA MEMORIA,54403428
6660 DATA EN LA MITAD DE LA MEMORIA,33393527
6670 DATA MUY ARRIBA EN LA MEMORIA,34383626
6680 DATA EN LA MATRIZ DE CARACTERES,35370025
6690 DATA EN UNA TABLA DE VECTORES ALEATORIOS,00000000
6700 DATA EN LA PARTE SUPERIOR DE LA MEMORIA CONTEMPLANDO UN BUS DESDE
     ARRIBA,39003735
6710 DATA EN LA MITAD DE LA MEMORIA,4003834
6720 DATA EN LA MEMORIA - HACIA EL ESTE HAY UNA PUERTA,41003933
6730 DATA EN LA PARTE INFERIOR DE LA MEMORIA,00004054
6740 DATA EN UN CORREDOR,00430031
6750 DATA EN UN CORREDOR,00440042
6760 DATA EN UN CORREDOR,00004543
6770 DATA EN EL REGISTRO DE DIRECCIONES,00004600
6780 DATA EN UN BUS DE 16 PISTAS,45004700
6790 DATA EN UN BUS DE 16 PISTAS,46004800
6800 DATA EN UN BUS DE 16 PISTAS,47004900
6810 DATA EN UN BUS DE 16 PISTAS SE VISLUMBRA
     HACIA EL OESTE UNA GRAN PUERTA 48005007
6820 DATA EN UN BUS DE 16 PISTAS,49005100
6830 DATA EN UN BUS DE 16 PISTAS,5005200
6840 DATA EN UN BUS DE 16 PISTAS,51000000
6850 DATA EN UN VECTOR A LA MEMORIA,00290012
6860 DATA EN LA PARTE INFERIOR DE LA MEMORIA,00413329
6870 REM ** DATOS SUMA DE CONTROL **
6880 DATA 100169,103973
8599 :
8600 REM **** S/R BUSCAR DIRECCION ****
8610 NN$=NN$+" ":LN=LEN(NN$):C=1
8620 FORI=1 TO LN
8630 IF MID$(NN$,I,1)<>" " THEN NEXT I:RETURN
8640 W$=MID$(NN$,C,I-C):C=I+1
8650 IF W$="NORTE" OR W$="ESTE" THEN NN$=W$:I=LN
8660 IF W$="SUR" OR W$="OESTE" THEN NN$=W$:I=LN
8670 NEXT I
8680 RETURN
```

mi computer

Editorial Delta, S.A.



LIBROS PARA TU MICROORDENADOR

18 JUEGOS DINÁMICOS PARA TU ZX SPECTRUM
por P. Monsaut, P.V.P. 650 Ptas.

En este libro se presenta una colección de 18 programas de juegos variados que combinan todas las posibilidades de su ordenador, color, gráficos, movimiento, etc. Además no sólo se limita a presentar juegos sino que aprovecha para mostrar algunos trucos y técnicas de programación.

18 JUEGOS DINÁMICOS PARA TU DRAGON 32
por P. Monsaut, P.V.P. 650 Ptas.

En este libro se presenta una colección de 18 programas de juegos variados que combinan todas las posibilidades de su ordenador, sonido, color, gráficos, movimiento, etc. Además no sólo se limita a presentar juegos sino que aprovecha para mostrar algunos trucos y técnicas de programación.

18 JUEGOS DINÁMICOS PARA TU COMMODORE 64
por P. Montsaut, P.V.P. 650 Ptas.

En este libro se presenta una colección de 18 programas de juegos variados que combinan todas las posibilidades de su ordenador, sonido, color, gráficos, movimiento, etc. Además no sólo se limita a presentar juegos sino que aprovecha para mostrar algunos trucos y técnicas de programación.

OTROS TITULOS

ZX SPECTRUM - QUÉ ES, PARA QUÉ SIRVE Y CÓMO SE USA
por Tim Langdell, P.V.P. 1.100 Ptas.

ZX SPECTRUM - APLICACIONES PRÁCTICAS PARA LA CASA Y LOS PEQUEÑOS NEGOCIOS
por Chris Callender, P.V.P. 870 Ptas.

PROFUNDIZANDO EN EL ZX SPECTRUM
por Dilwyn Jones, P.V.P. 1.300 Ptas.

CÓMO CREAR TUS JUEGOS SPECTRUM
por R. Rovira, P.V.P 750 Ptas.

DRAGON - QUÉ ES, PARA QUÉ SIRVE, CÓMO SE USA
por Ian Sinclair, P.V.P. 1.300 Ptas.

COMMODORE 64 - QUÉ ES, PARA QUÉ SIRVE Y CÓMO SE USA
por D. Ellershaw y P. Schofield, P.V.P. 950 Ptas.

COMMODORE 64, APLICACIONES PRÁCTICAS PARA LA CASA Y LOS PEQUEÑOS NEGOCIOS
por Chris Callender, P.V.P. 830 Ptas.

INTRODUCCIÓN AL MSX
por Vanryb y Politis, P.V.P. 1.100 Ptas.

DICCIONARIO MICROINFORMÁTICO
por R. Tapias, P.V.P. 990 Ptas.



EDITORIAL NORAY, S.A.
San Gervasio de
Cassolas, 79
08022 Barcelona
ESPAÑA
Tel. (93) 211 11 46